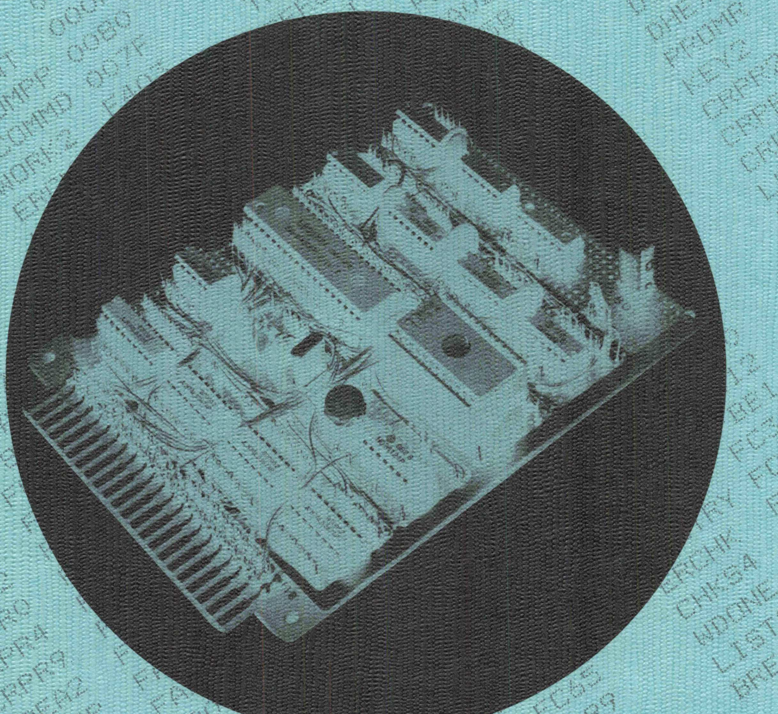
CP/M手作りマイコン

安居院猛・中嶋正之・清弘智昭



産報出版

CP/M 手作りマイコン

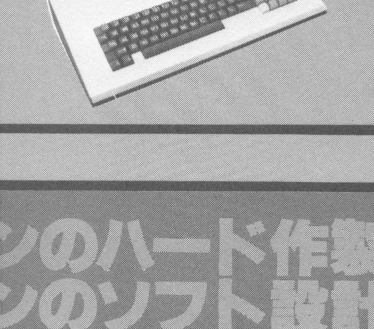
安居院猛・中嶋正之・清弘智昭

産報出版

第1部　CP/Mマイコンのハード作製
第2部　CP/Mマイコンのソフト設計
第3部　付　録



第1部　CP/Mマイコンのハード作製
第2部　CP/Mマイコンのソフト設計
第3部　付　録



まえがき

本書は，CP／M＊が搭載されたマイコンを，安価に，かつ短期間で入手することを目的として書かれたものです。現在，CP／Mが大流行しており，マイクロプロセッサを用いたシステムを開発する際には，必要不可欠なシステムとなりつつあります。

この手作りコンピュータは，筆者らが自ら基本設計を行ない，かつ作製，そして基本ソフトの開発を行なったものであり，冬休みを利用して約1カ月で完成したものです。いまでは，この手作りコンピュータも，1台では足りず，2台目が完成し，CP／Mの下でソフトウエア開発の際に大活躍をしています。

どちらかといえば，ソフトウエアが専門である筆者らが，短期間で作り上げた装置のため，現在ではハードウエアの点で多少不満足な点があります。しかし，60KメモリサイズのCP／Mという便利なシステムが導入されていることで，ソフトウエアの面からは“手作り”という感じがまったくしない高価なマイコンが完成したという実感がします。

ハードウエアの点からいえば，将来，倍密のフロッピーディスクが使えるように，前もってDMAが可能となる設計にすべきであったとか，フロッピーディスクコントロールICは，FD 1771よりも使いやすい5V単一電源のFD用ICにすべきであったかもしれないなど，反省点は多少残ります。しかしながら，インターフェースICの最近の進歩はめざましく，たえず最新のものを追求していたら，いつまでたっても完成しかねません。

したがって，多少古くとも広く使われ，実績のある標準的なICを使用するのも得策であると考えられます。その点から考えて，この手作りコンピュータの設計は，ほぼベストであると考えています。

なお，これらの改良や，フロッピーディスクを2台以上取り付け可能にするなどの設計変更は，簡単に行なえますので，読者の強い希望があれば，雑誌「電子科学」(産報出版)の誌面上ででも，改造説明を掲載することを考えています。

この手作りコンピュータが1台でも多く作製され，CP／Mの下で，ソフトウエア開発に役立つことを願うものです。

おわりに，本書の出版に関して，お世話になった産報出版の藤林　明氏に厚く感謝します。

昭和57年6月

著者記す

＊ CP／M(Control Program／Monitor) はディジタル・リサーチ社の登録商標です。

CP／M
手作りマイコン

目　次

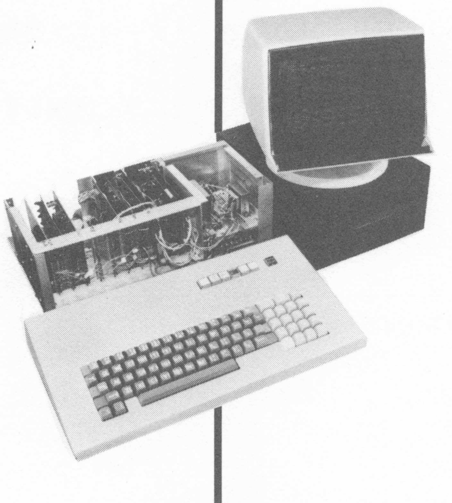
写真提供＝摂津金属工業㈱
(図5.1)

第1部　CP／Mマイコン

第1章　はじめに

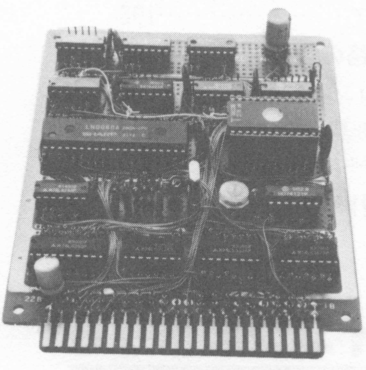
第2章　ハードウエア入門

第3章　マイクロプロセッサ



のハード作製

第4章　基本回路の作り方



第5章　マイコン本体の作り方

第6章　PROMライタの作り方

第7章　基本回路のテストとモニタの作り方

第2部 CP/Mマイコンのソフト設計

第8章 フロッピーディスク制御回路の作り方

第9章 CP/Mマイコンの作り方

第3部 付 録

第1部
CP/Mマイコンのハード作製
第2部
CP/Mマイコンのソフト設計
第3部　付　録

第1部　CP/Mマイコンのハード作製
第2部　CP/Mマイコンのソフト設計
第3部　付　録



第1部　CP/Mマイコンのハード作製
第2部　CP/Mマイコンのソフト設計
第3部　付　録



第1部第1章

はじめに 1

はじめにこの本の構成と，これから作製する手作りコンピュータ，CP/Mマイコンについて概略を説明しておきます。

1.1 本書の構成について

マイクロコンピュータ，略してマイコンが大変普及し，テレビでさえ盛んに宣伝されるようになってきました。まさに，マイコンブーム到来の感があります。書店では，マイコン関係の本が多く並べられ，小学生から高年齢層にわたるまで興味がもたれています。

しかし，安価になってきているとはいえ，現在出まわっているマイコンは，CPU本体は安価でもフロッピーディスク，プリンタやカラーディスプレイなどの各種入出力装置を付属させると，かなり高価なものとなります。マイコンを機械や計測器などのコントローラとして利用する場合は，購入したマイコンに関するハードウエアの構成に関する知識を習得しなければなりません。

また，マイコンを買い，メーカの指定通りの動作をすることは理解しても，それだけではマイコンを自由に使いこなせたという満足感が得られないという感想をもっている人も少なくありません。

このような既製のマイコンにあきたらず，自分でオリジナルのマイコンを安価に製作してみたいと多くの方々が思っていると考えられます。そこで，本書はハードウエアの知識があまりない人およびマイコンになじみのない人を対象として，もっとも基礎的な段階から手作りのCP／Mマイコンを作る方法を，ていねいに解説しました。

本書は**マイコンを作ること**に目的を絞っており，各章には，手作りコンピュータを作製するための具体的な作業内容を詳しく記述してあります。いままでの既製のマイコンの利用にはあきたらなくなった人や，自分でも1台実用可能なマイコンを入手したいと考えていた読者は，ぜひこの機会に“CP／M手作りマイコン”にチャレンジして下さい。

■初心者を対象

本書は，読者層をハードウエアの専門家を対象にせず，あくまでもソフトウエア技術者，すなわちハードウエアの経験の少ない人々を対象としています。そのため，専門家が当然と思っているようなことでも，素人には気がつかない重要なポイントを中心に記述してあります。

また，この手作りコンピュータの作製に当たっては，実際に設計図面を，ハード知識がない初心者に与え，配線および動作チェックを経験させ，その際に見いだされた注意点を極力きめ細かく記述するように努めました。こういった意味でも，初心者のためのハードウエア設計の手引書ともいえる構成となっています。

■製作するCP／Mマイコンの特徴

製作するマイコンの構成は，ICが20個程度のるハガキ大のユニバーサルボードを5枚利用し，各ボード単位で独立した機能となっています。すなわち，1枚ごとに着実に動作をチェックしながら作業を進めることにより終局的には全システムが完動するようにしました。そのため，各基板ごとの動作が習得しやすい形式となっています。

筆者らの最初の構成では，大きなユニバーサルボード1枚に，全回路を搭載することを考えていました。しかし，動作チェックのしやすさ，階層的な積み上げによってマイコンを作ることを考慮した場合，作業別分割方式がベストであると考えました。基板の枚数が多くなることは，バスラインをそれだけ引きまわすことになり，雑音に弱くなるという欠点があります。しかし，それにもまして，1枚の基板で各動作ごとに必要とするすべての回路が搭載されているというメリットはきわめて大きいと考えられます。

1.2 製作するマイコンの概要

第1章　はじめに

製作するマイコンは**図1.1**に示すように，全部で5枚の基板から構成されています。5枚の基板が完動すれば，標準フロッピーディスクおよびプリンタが付属したCP／Mマイコンが利用できることになります。

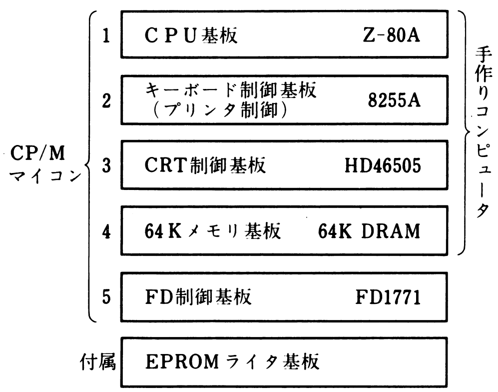


図1.1　手作りマイコンの基板構成

しかし，CP／Mマイコンまでは入り用のない場合には，フロッピーディスクコントロール基板は必要としません。残り4枚だけでも64Kバイトメモリのマイコンが完成します。したがって，本書では第1部の第7章までが基本的マイコン部分で，回路の製作について詳しく述べてあります。前半だけでもマイコンの手作り方法の習得には役立つものと考えています。

数年前に比べて安価になったとはいえ，フロッピーディスク装置は，まだまだ手軽に入手できる価格ではありません。そのような読者のためにも，本書ではフロッピーディスクを購入しなくてもマイコンとして十分利用可能なように考慮しました。

また，EPROMライタをもっていない読者のために，手動書込みと手作りマイコンによる自動書込みが兼用できるEPROMライタ基板の製作と操作方法についても述べてあります。

■インターフェースICの有効利用

製作するマイコンの特徴は，周辺装置用のICを最大限利用している点です。CRTのコントロールには，日立のIC（HD 46505）を用い，フロッピーディスクコントロールにはコントロール用ICのFD 1771（ウエスタンディジタル社製）を用いました。

このようなインターフェース用ICを利用したことにより，より少ない作業時間で，かつハガキ大の汎用ユニバーサル基板上に各種インターフェース回路を実現することが可能となりました。専用ICを利用することは，これからも便利で複雑な動作をする専用コントロールICが逐次市販されることが予想されますので，これらのICを確実に利用するための良い練習になると考えています。

■ 64 KDRAMの利用

メモリが急速に安価に出まわるようになりましたので，ここで作製するマイコンでは64Kの

ダイナミックRAM（Random Access Memory）を8個利用し，64 Kバイトという豊富なメモリを搭載させることにしました。

本書では，手作りコンピュータを作ることが目的ですが，このDRAM部分の有効利用として，手持ちのマイコンに対して64 Kバイトの外付けメモリを安価に実現させせようと考えていた読者にも，掲載した回路は役立つと考えられます。

1.3 本書の効果的な読み方

第1章　はじめに

本書は，はじめてマイクロコンピュータを手にした初心者を対称として，実際にマイコンを作るための“すべて”について述べてあります。すでにマイコンのハードウエア知識をもっている人は，第1～第3章は軽く読み飛ばし，第4章の作製から開始して下さい。以下，本書の効果的な読み方について示します。

■EPROMライタをもっている人

2716または2732用のEPROMライタをもっている人は，第6章の“PROMライタの作り方”および第7章2項の“自動EPROMライタ回路のテスト方法”と3項の“基本モニタプログラムの作り方”は不要です。この場合，図1.1に示す5枚のCP／Mマイコン基本回路が完成したら，すぐ**付録6**に示すダンプリストをCP／Mモニタプログラムとして書き込み，フロッピーディスク制御基板が完成したら，CP／Mの実装を試みて下さい。

また，2716用のEPROMライタをもっている人は，キャラクタジェネレータとして安価に購入できる2716を利用することをおすすめします。

■CP／Mシステムを利用できる人

すでにCP／Mシステム（任意のメモリサイズ，バージョンでも可，またターベル版でもディジタル・リサーチ版でもかまいません）を利用できる場合は，この手作りコンピュータに60 K CP／Mを実装する作業がきわめて簡単になります。利用できるCP／Mを用いて，60 KメモリサイズのBOOTプログラムとCP／M本体のプログラムを作り（この作業は簡単に行なえます），**付録7**にアセンブラリストを示す手作りコンピュータ用のBIOSのHEXファイルを組み込めば，60 KメモリサイズのCP／Mマイコンが完成します。

CP／Mの実装は，現在のところアマチュアにおける**手作り**マイコンの最高の到達点です。しかし，CP／Mを実装しなくとも，コンピュータとして，次のような利用方法があります。

①　各種制御装置としての利用

キーボード入力を各種外部装置からのセンサ入力として利用し，Bポートを外部装置への制御信号として利用すれば，基板が2枚だけで各種制御装置として利用できます（**図1.2**参照）。この試作は，マイクロプロセッサを知り，その利用方法を習得する恰好の作業となるでしょう。温度センサとパワリレーを組み合わせて，室内温度制御装置でも作ってみたらいかかでしょうか。

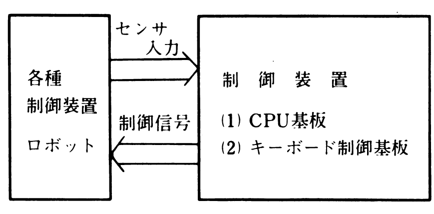


図1.2　各種制御装置のコントローラ

②　外付け用拡張64 Kメモリ装置

手持ちのコンピュータに対する拡張メモリとして利用できます。この場合は，**図1.3**に示すようにキーボード制御基板が並列のデータ入力となり，ポートBをデータ出力として利用し，他はCPU基板と64 Kバイトメモリ基板だけで完全に動きます。この64 Kメモリ装置の試作は，安価に市販されている64 KダイナミックRAM

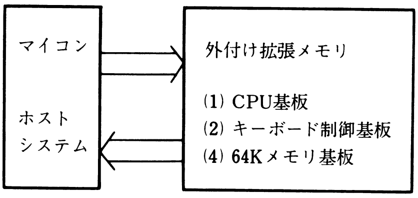


図1.3　外付け拡張メモリの試作

の使い方の学習となります。

③　キャラクタディスプレイ装置

①のCPU基板と②キーボード制御基板，③CRT制御基板を用いることにより，計算機の端末用のキャラクタディスプレイ装置として利用できます。このキャラクタディスプレイの試作は，ディスプレイコントローラ用ICの使用に関する最適な学習となるはずです。

また，一部設計を変更すれば，256×256ドットのグラフィックディスプレイとしても利用できます（**図1.4**）。これらの試作は，きわめて安価に行なえますので，はじめての自信のない人でもマイコンの第一歩としてぜひ試みて下さい。

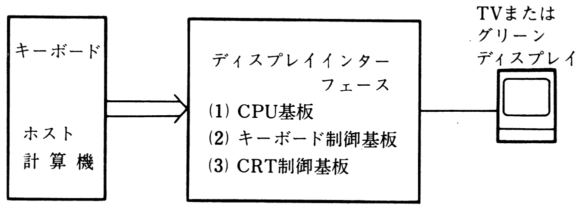


図1.4　キャラクタディスプレイ装置

④　標準フロッピーディスクインターフェース

手持ちのマイコンにぜひ標準フロッピーディスクを搭載してみたいと考えている人は，**図1.5**に示す構成で簡単に実現します。もちろんFDコントローラだけの利用であれば，64Kメモリは不必要なので，64K DRAM基板は2114のスタティックRAMを2個搭載した1Kバイトメモリ構成ですみます。

⑤　RS232Cインテリジェントターミナルとしての利用

その他，電話回線などを通じたターミナル用としても利用できます(**図1.6**参照)。この場合は，新たに26ページに示すRS232C用インターフェース基板を追加すれば良いことになります。

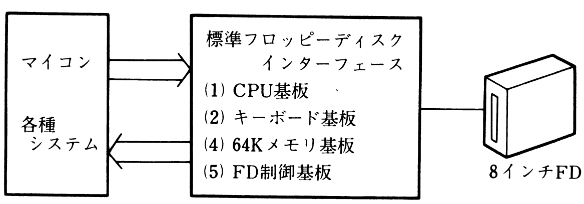


図1.5　標準フロッピーディスクインターフェース

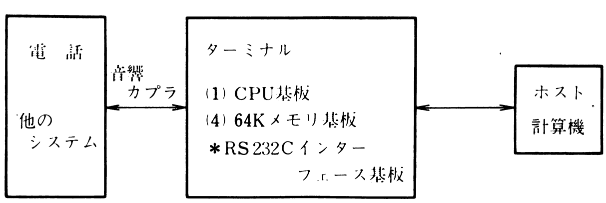


図1.6　ターミナル装置

第1部第2章

ハードウエア入門 2

ここでは，手作りコンピュータのハードウエア実装に関して必要となる基礎知識について，述べます。

2.1 ハードウエアの基礎知識

まずはじめに，手作りコンピュータを作製するときに必要な工具および基礎知識について説明します。

2.1.1 ラッピングについて

■ラッピング

手作りコンピュータを製作するときの配線に必要な最小限の工具購入リストを**表2.1**に示します。この表の中で，はんだゴテは絶対必要な工具であり，多くの読者はすでに購入ずみであると考えられます。これから購入する場合は，10W～20W程度のIC専用のコテを購入して下さい。

手巻きラッパとワイヤストリッパは，すべてはんだ付けで配線を行なう人には必ずしも必要な工具ではありません。しかし，はんだ付けは初心者には大変難しい作業であり，一見接続しているように見え，また導通していても，軽い機械的な振動などにより離れてしまい，いわゆる"イモハン"が生じやすくなります。

そのため，はんだ付けの良否で，信頼性が左右され，正常に動作したり誤動作したりする装置となる危険性が生じやすくなります。そこで，初心者は極力はんだ接続の個数を少なくすることが信頼性の高いコンピュータを作る条件となります。ラッピングによる配線をおすすめします。

表2.1　配線に必要な工具

工具名	仕様	備考
はんだゴテ	10W～20W	
ラッパ	線太　0.25mm（30m）	アンラッパつきの手巻きラッパ WSV－30M（O.K.M.&T CORP）約1,800円 電動ラッパ
アンラッパ	線太　0.25mm	アンラッパつき手巻きラッパを購入した人は不必要
ラッピング線用のワイヤストリッパ	線太　0.25mm	例えば，FREMONT製（規格014）
ピンセット	先の細いもの	

ラッピングとは**図 2.1** に示すように，ラッピングピンのまわりに細い線を 6 ～10 回巻きつけて配線を行なう方式です。特徴としては，初心者にも手軽に扱え，正しい取扱い方をする限り，信頼性のある回路配線が行なえます。

また，誤配線のときの処理も，プリント板を損傷することなく比較的簡単に行なえます。

はんだ付けをした場合，誤配線を直すのに熱を加えることを繰り返すと，プリント板の銅板パターンが板から剝離し，信頼性の低下を招くことになります。

図 2,1 に示すように，ラッピングは被覆部分を 1 回以上巻きつけてあるので，比較的線の強度が保てるため，断線しにくいという特徴があります。このラッピング線を巻きつけるときに必要な工具がラッパです。

電動式のラッパが普及しており,大変便利なものですが,簡単な手巻き式のラッパでも十分実用になります。手巻き式ラッパは，安価である上に持ち運びが簡単で,電源が不必要なため,どこでもラッピングが行なえるという便利な点があります。

通常,手巻きラッパは,ラッパの逆側がラッピング巻線をはずすアンラッパとして使用できるため,別にアンラッパを購入する必要がありません。したがって,ぜひ 1 本揃えておきたい工具です。また,配線を整えるためにピンセットを用意します（68 ページ参照）。

その他に必要な工具は，ラッピング専用のワイヤストリッパです。これはラッピング線のビニル被覆を取り除くために必要な工具です。

実際の配線作業おいて，線の被覆除去が正確に行なえるか否かは重要です。ワイヤストリッパだけは耐久性があり，使用しやすい工具を用意して下さい。**図 2.2** は，多少高価ですが，使用しやすいストリッパの一例です。

ラッピングの方法は,図 2.1 に示したように,すきまなく **10 巻**程度ラセン状に巻きつけることが必要です。ラセンの間隔があいたり，線と線が重なったり,巻かれる回数が少なかったり,ビニル被覆部分が最低 1 回以上巻かれていない場合は，悪いラッピングです。この場合は，必ずラッピングをやり直します。

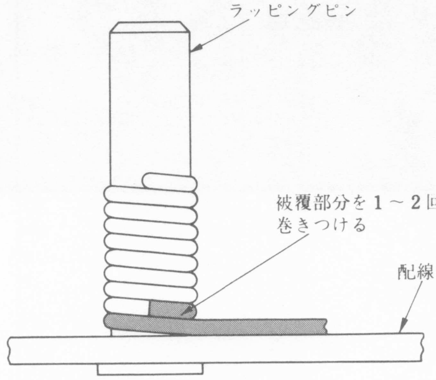


図2.1　ラッピング

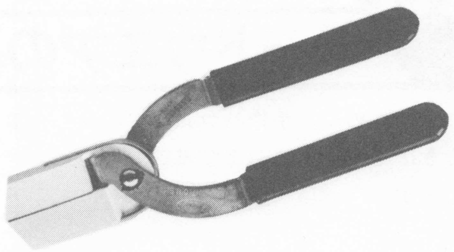


図2.2　ワイヤストリッパ (FREMONT社製，規格014)

■誤配線の処理

誤って配線した場合は，アンラッパを用いてラッピング線を取り除きます。その際，**図 2.3(a)** に示すように，下段のラッピングを取り除くためには，上のラッピング線から取り除かなければなりません。したがって数本分のラッピング線を除去しなければならない場合があります。

この場合の対策として，新しく配線するピンの巻き線部分に余裕があるときは，図 2.3(b)に示すように，根元から誤った線を切断してしまう方法があります。また，図 2.3(c)に示すように，1 本の線をはずすためにその影響が伝播し，数本のラッピング線を除去しなければならない場合が生じます。

この伝播を防ぐには，図 2.3(d)に示すように，ラッピング位置の高さを揃えるように注意します。

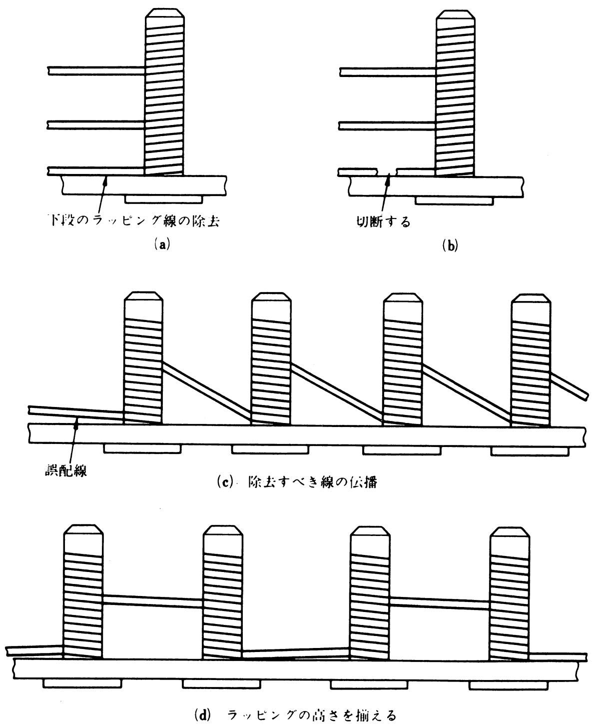


図2.3　誤配線の処理方法

2.1.2　配線に必要な部品

マイコンの作製に当たって必要な工具について述べましたが，次に配線に必要な電気部品について説明します。必要とする主要な部品の一覧を**表 2.2** に示します。

■配線基板

配線を行なう基板は，市販のユニバーサルボードを使用します。1枚に最低限 20 個以上のTTLが実装可能なボードを必要とします。実装効率（44 ピン，サイズ 115×155 ㎜で 20 個実装可能）および価格の点から，**図 2.4** に示す亜土電子製のモデル 801 がいいでしょう。以下配線は，このモデル 801 を使用することにして説明

表2.2　配線に必要な部品

部　品　名	数　量	備　考
ユニバーサルボード	6 枚	亜土電子製モデル801(ベーク板)，約600円
ラッピングピン	約 1,000本	ラッピングピンつきのソケットを購入する人には不要
IC ソケット		14ピン～40ピン
コネクタ　44ピン	9 本	4 ㎜ピッチ，KEL　1150－044－009
コネクタ　22ピン	1 本.	4 ㎜ピッチ
ラッピング線	200 m	0.25㎜
スズメッキ線	少々	
I C	**付録 8** 参照	必ず新品または動作チェックずみのICを用意すること

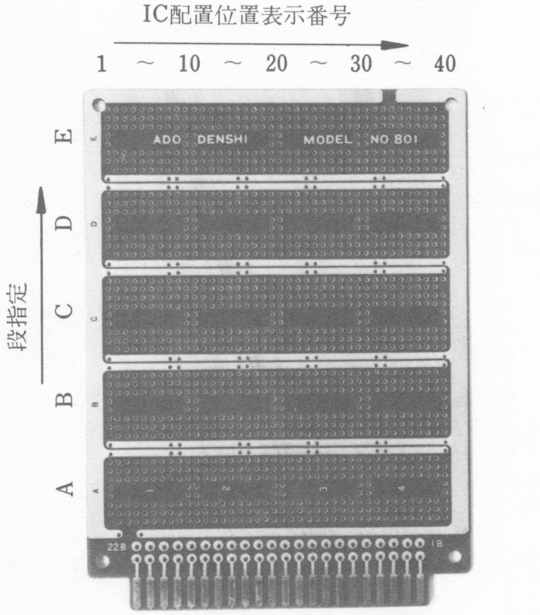


図2.4　ユニバーサルボード（亜土電子，モデル801）

します。

モデル801は，A～Eの5段構成で，各段に20ピンのTTL-ICを4個実装できます。そして，Cの段が幅広く構成されており，24ピンや40ピンの幅広のICが実装可能となっています。

またモデル801が使いやすい点としては，TTL-ICの電源の配線がしやすい構成であるということです。すなわち，コネクタ端子番号A1とB1をGNDに，またA22とB22に5Vを加えることにより，TTL-ICの電源のピン位置の近くまで供給されることになります。

■ICソケット

使用するICはすべて直接固定することなく**必ずICソケットを使用**する必要があります。

ICソケットには，はんだ付け用と直接ソケットの足にラッピングが可能となっているラッピング用があります。このラッピング用ソケットは，ラッピングピンを立てる必要がないので，その点便利です。したがって，ラッピング用ソケットを購入することが望ましいのですが，安価に出まわっているはんだ付け用ソケットでラッピングピンを立てる方法もあります。

■ラッピング線

ラッピング線は，0.25(0.26)mmφを使用します。色は1種類でもいいですが，数色取り揃えた場合は，アドレスラインを赤色，データラインを黄色，その他の配線を橙色というように決めておけば，後の配線チェックも多少わかりやすくなるという有利さがあります。またラッピング線は，ストリッパで被覆を除去しやすいことが望ましいので，例えば潤工社製のジュンフロンETFE電線（CSMW用）などが使いやすいでしょう。

■ラッピングピン

プリント板モデル801を利用した場合，コネクタ端子から配線を引き出す必要があります。また，ラッピング用のICソケットを使用しなかった場合は，ICソケット部分に対しラッピングをするのにラッピングピンを使用します。

このラッピングピンは，配線をする必要がある部分にはすべて立てます。ピンの位置はICソケットからもっとも離れたところとします。

■はんだ付けの仕方

はんだ付け用のICソケットを用いた場合，ラッピングピンをつけるときや抵抗，コンデンサ，ビニル線をつける際は，はんだ付け作業を必要とします。はんだ付けの作業は，はんだゴテの使いやすさにより，良，不良が左右されます。

コテ先をクリーンな状態にしておくことがもっとも重要です。そのため湿ったスポンジつきのコテ台を用意しておいた方がいいでしょう。

はんだ付け作業の方法としては，はじめにコテ先をプリント板パターンに接触させ，次に，ヤニ入りハンダをつけて十分ハンダをしみこませます。ハンダが表面張力によりまわりに広がり，流れる程度まで熱を加え続けます。はんだゴテを離したとき，光り輝くような状態になればいいでしょう。

■半導体部品

使用するICの一覧を**付録8**に示します。この場合，必ず新品を用意するか，または完全に動作チェックずみのICを使用することが望まし

いでしょう。

手作りコンピュータでの誤動作または動作しない原因としては，配線図が正しい限り，誤配線またはIC不良の場合が主です。

誤配線は，十分チェックすることにより防ぐことができますので，IC不良がもっとも扱いにくいでしょう。誤動作の原因追求のための無駄なエネルギーを費やさないためにも，必ず完動するIC（新品）を使用しなければなりません。

2.1.3 配線の検査

配線が終了してから，完動するまでに必要な検査手順について述べます。必要とする主な装置を**表 2.3** に示します。

表2.3 動作チェックに必要とする装置

装置	用途
テスタ	導通チェック用
拡張ボード（ゲタ）	44ピン配線基板用
シンクロスコープ	4 MHz 観測可能のもの

■配線ずみの明示

配線は図面に従ってラッピングを施してゆきます。このとき，配線を行なう図面のコピーと細字用のマジックペンを用意する必要があります。これらは，配線ずみを明示するもので，**1本の配線を終了したとき図面上の配線部分をマジックで色づける**ことにより，配線の脱落が未然に防げることになります。

1本のラッピングごとにマジックで色づけする作業は，多少めんどうに感じられるかもしれませんが，逆に確実に動作する電子回路を短時間で作り上げる近道となります。必ず励行するように心がけて下さい。

■配線チェック

配線は，各ICの5VおよびGNDの電源配線から行ないます。配線図中にはTTL-ICの電源ピン位置はとくに示してありませんが，14ピンでは7番がGNDで，14番が＋5V，16ピンでは8番がGNDで，16番が＋5Vにほぼ定められています。

配線がすべて終了したら，次にICをのせる前に，配線ボードの配線チェックを行ないます。

《配線チェッカの作り方》

用意するもの（**付図 2.1** 参照）

① 名刺入れ用プラスチック箱または，それに類する小さなプラスチックボックス

② ブザー1.5Vまたは3V用（3V以上のブザーは使用しないこと）。このブザーは電池をつなげば音が出るものを使って下さい。圧電ブザーと称して，振動板単体のものが市販されていますが，これでは電池をつないでも音が出ませんので，注意する必要があります。

③ 単3乾電池ボックス（3Vブザーの場合は2列用）

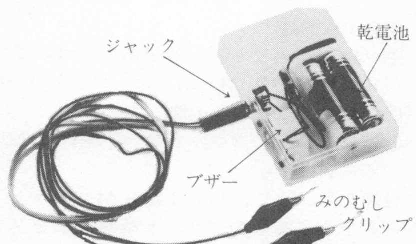
④ オーディオ用ジャック1組

⑤ みのむしクリップ赤と黒各1個

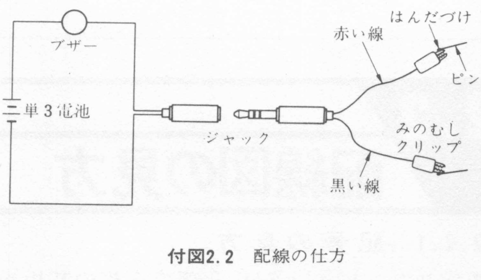
⑥ コード。赤と黒各40 cm程度

付図 2.2 に示すように配線をします。すなわち，みのむしクリップをつけた2端子を導通するとブザーが鳴るようにすればいいことになります。

この場合，電池の＋側は赤いコードと赤のみのむしクリップを使用し，アース側は黒いコードと黒のみのむしクリップを使用するといいでしょう。また，みのむしクリップの先にはピン（例えばラッピングピン）をはんだ付けしておくとICチェックが行ないやすくなります。



付図 2.1 配線チェッカのできあがり



付図2.2 配線の仕方

この場合，配線図に従って1本1本図面通りに配線が行なわれているか否かの導通テストをします。この導通テストは，手持ちのテスタの抵抗値を測定することにより行なえます。

テスタをもってない読者のために，千円未満で作れる簡単な配線チェッカの作製を紹介しましょう。前ページの下囲みを見て下さい。

このチェッカは，音により配線の有無を検査します。テスタの針を見るという動作が不必要なので，とても便利です。短時間で作製でき，テスタをもっている人でもあると便利なものです。

導通テストは，大変やっかいな作業ですが，従来の経験から誤配線は必ずありますので，励行することをすすめます。とくに，ICの電源部分は，正しく配線が施されているかを検査する必要があります。電源の誤配線は，ICの損傷を招きますので，必ず実行するように心がけて下さい。

■IC搭載後のチェック

配線チェックが終了し，誤りなく配線されていることを確認したら，ICを搭載し，電圧を加え，正しく動作することの確認を行ないます。

各基板ごとの動作のチェック方法は，第5章で個別に述べます。したがって，ここではその場合の注意すべき点についてふれます。

第一に，正しく動作しない場合，再び配線チェックなどが必要になります。このチェックの際，ICを搭載したままでもいいですが，必ず電源を切って行なうことが大切です。電源投入のまま，みのむしクリップの先端をICピンに接触させると，誤って他のピンに接触したとき，ICを破壊する恐れがあります。

この事故を未然に防ぐためにも，**電源を必ずOFFにした状態でチェックすること**が重要です。各基板をコネクタから抜き差しする場合も，電源をOFFにしておかなければなりません。

また，電源を入れたままシンクロスコープなどで波形を見ながら調整する場合がありますが，そのときに必要な道具として，俗称"ゲタ"と呼ばれる拡張コネクタを準備すればいいでしょう。チェックしたい基板を手元に配置することができ，大変便利なものです。下に作り方を示しましたので，用意して下さい。

《拡張コネクタの作り方》

用意する部品

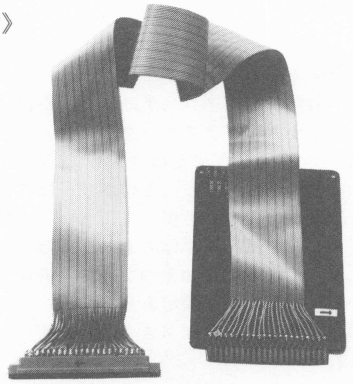
① 44ピンのソケット端子部分がついているプリント板1枚（使い古したものでもいい）

② 44ピンソケット1個

③ 44本以上のフラットケーブル60cm程度（22本を2組でもいい）

付図2.3に示すように，1から44のソケット端子から対応する番号のプリント板端子間をフラットケーブルでつなぎます。

その場合1番ピンの方向を必ずマジックで目印をつけることが大切です。これは逆に挿入することを防止するために必要です。



付図2.3　44ピン拡張コネクタの作製例

2.2 配線図の見方

2.2.1　記号の見方

手作りコンピュータは，全部で5枚の基板から構成されており，各基板ごとに異なる機能をもち，基板ごとに1枚の配線図が対応します。

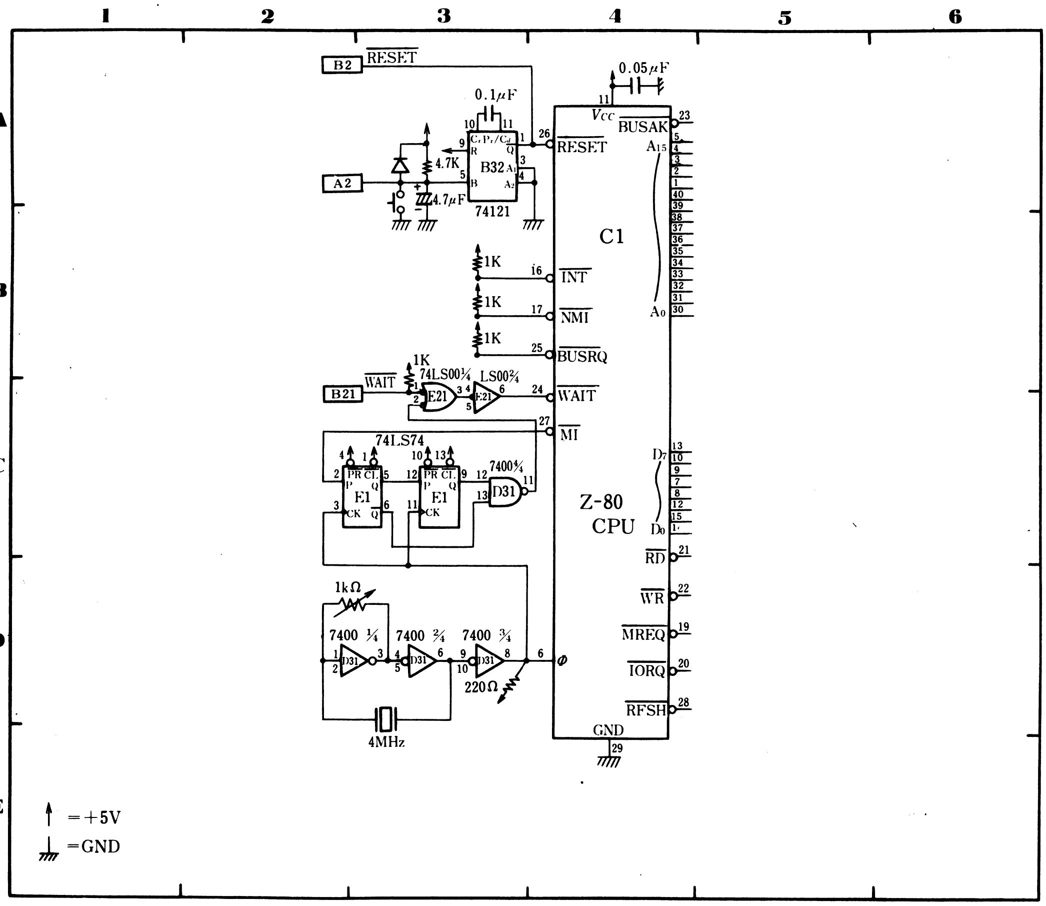


図2.5 配線図の例

各基板ごとの各ICの動作の詳しい説明を第4章以降で行ないますので，本節では配線図で使われる記号や番号の意味について述べることにします。

■図面内位置の表示

図2.5に示すように，配線図の外側は枠取りが施してあり，上側には左から右へ**1**，**2**，**3**……**8**，という数字と，左側には，上から下へ**A**，**B**，**C**……**F**と英字が記入してあります。

これは，地図の経緯線と同様の役割をしており，図面内の位置を大まかに示すものです。例えば，“CPUの4MHzクロックを発生させる回路は，CPU基板の**3D**の位置にある”というように説明します。

本書では，回路の位置の表示のみに利用しますが，その他通常は，複雑な図面における線の行方の表示，異なる基板への配線位置の表示，2枚にわたる配線図間の線の位置表示などに利用できます。

■コネクタ端子の番号

図2.5に示す配線図において，**2A**の位置に□で囲った記号があります。この記号は，配線基板のソケット部分の番号を示すものです。

モデル801ユニバーサル基板は，ICがつく部品側に，B1からB22までの22本の端子，また，ソケットのはんだ付けを行なう裏側に，A1からA22の22本の合計44本の端子が出ています。この端子に4㎜ピッチの44ピンソケットが接続されることになります。

また，ソケット端子記号─□に入る線上には，その端子の信号の意味を記入してあります。例えば，本図面上の**2A**に示したソケット端子[B 2]はCPU Z-80への$\overline{\text{RESET}}$信号であることを示しています。

■IC名

配線図上のIC素子記号の付近には，使用するIC名が記入されています。例えば，**図2.6**の回路におけるNOT素子は，7400(正式にはSN 7400)を使用することが明記されています。またIC名7400の次に示す1/4，2/4，3/4は，1個の7400には，4つのNAND回路があり，その4つのNAND回路中の1番目（ピン番号1，2，3），2番目（ピン番号4，5，6），3番目（ピン番号8，9，10)をそれぞれ利用することを明示したものです。

その他の記号としては，矢印↑で5V，⏚でGNDを示すことにします。例えば図2.5の図面中のZ-80 CPUは29番ピンをGNDへ，また11番ピンを5Vへ接続することを示しています。

■IC位置表示

図2.6に示す発振回路は─▷─で示す3つのNOT素子回路から構成されており，─▷─内にD 31と記入されています。同様に，図面内の各論理素子シンボル内には，必ずA～Eではじまる英字と，2桁以内の数字が記入されています。

これは，配線基板の板面上の対応するICが配置されている場所を示すものです。この位置記号を用いることにより，配線図から直接対応するICを見つけることができます。

はじめの英字は，図2.5に示す配線基板の左側に明記されている英字A～Eに対応しています。もっともソケットに近い段をAとし，もっとも上部の段がEとなっています。すなわち，英字はソケットからの段を示していることになります。

次に数字ですが，これはICの1番ピンの位置を示すものです。図2.5に示したプリント板の各段は，40列の穴があいており，英字が記入されている方から1番，2番……と定めます。

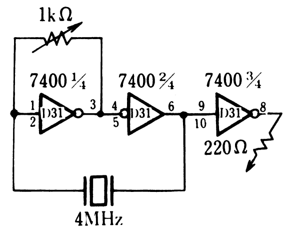


図2.6　図2.5の配線図内の**3 D**の位置にある発振回路

例えば，4 MHz発振器にはD 31と記入されていますので，用いるTTL-IC 7400の1番ピンがD 31の位置にくるように配置することになります。同様に，図2.5のZ-80 CPUには，C 1と記入されていますので，Cの段のもっとも左側に配置することになります。

■配線ピン番号の表示

各ICのピンにラッピングを施す際のピン番号が，各IC素子の入出力線上に記入されています。例えば，図2.6の発振回路においては，"D 31に配置されたIC 7400のピン1，2は，1 kΩのボリュームと4 MHz水晶へ，またピン番号3は1 kΩのボリュームの片方と，4と5番のピン番号を接続させる"ということを示しています。

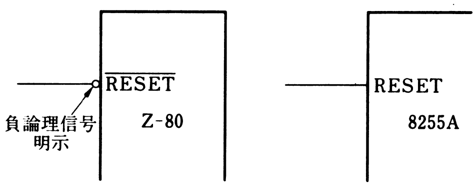
2.2.2　負論理について

図2.5に示すCPU基板の配線図には，IC素子記号に入る線の入口に小さな丸印が多数存在することに気がつかれたと思います。これは，対応する信号が負論理信号であることを示す大切な記号です。

そもそもTTLによるディジタル回路は，使用する信号が3V以上の電圧が対応する"High"状態と，2.4V以下の"Low"状態の2値から構成されており，各信号は必ず"High"，"Low"のどちらかの値を示すことになります。

このとき，各信号が"High"**のときに動作する場合を正論理**，"Low"**のとき動作する場合を負論理**とよんでいます。

例えば，リセット（初期状態にする信号）をとりあげてみましょう。CPU基板上のZ-80 CPUのリセット信号は，図面上**4A**の位置に書



(a) 負論理による$\overline{RESET}$ (b) 正論理によるRESET

図2.7 負論理リセット信号$\overline{RESET}$と正論理リセット信号RESETの明示方法

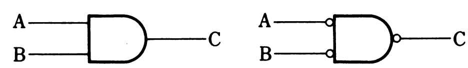
表2.4 AND 回路の真理値表

(a) AND 回路 (正論理)
C＝A・B

A	B	C
Low	Low	Low
Low	High	Low
High	Low	Low
High	High	High

(b) AND 回路 (負論理)
C＝A・B

A	B	C
Low	Low	Low
Low	High	High
High	Low	High
High	High	High



(a) 正論理のAND回路 (b) 負論理のAND回路

図2.8 AND回路のシンボル

表2.5 OR 回路の真理値表

(a) O R回路 (正論理)
C＝A＋B

A	B	C
Low	Low	Low
High	Low	High
Low	High	High
High	High	High

(b) O R回路 (負論理)
C＝A＋B

A	B	C
Low	Low	Low
High	Low	Low
Low	High	Low
High	High	High

かれており，負論理です。すなわち，2.4 V以下の電圧が加わっているときリセット状態になることを示すため，図面上で**図 2.7 (a)** に示すような○印がつけられ，かつ$\overline{RESET}$と——がつけられています。

それに対し，後述する**図 4.6** に示すインターフェースIC 8255 Aのリセット信号は正論理，すなわち 3 V以上の電圧が加わることにより，リセット状態となります。したがって○印がついていないことになります。

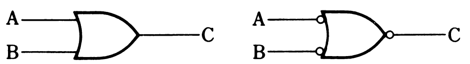
TTL-ICでは，入力端子が開放の状態では通常 "High" を示します。そのため，誤っても誤動作を少なくするように，マイクロプロセッサの多くの信号は，負論理信号から構成されます。例えば，**図 4.2** に示したZ-80 から出るリード信号$\overline{RD}$，ライト信号$\overline{WR}$，メモリリクエスト信号$\overline{MREQ}$などは，すべての信号が負論理となっています。

■負論理

正論理におけるAND回路は，よく知られており，**表 2.4 (a)** にその真理値表を示します。すなわち，2つの入力信号A，Bが"High"のときのみ出力Cが"High"となる回路です。このAND回路は**図 2.8(a)**の記号で示されます。

それに対し，負論理におけるAND回路は，2つの入力信号が "Low" のときにのみ出力が "Low" となる回路として構成されます。すなわち真理値表は，表 2.4(b)に示す構成となります。またその配線図上での記号は，図 2.8(b)のように示され，同図(a)の正論理とは異なり，2つの入力端子に○印がつけられます。

この場合注意すべき点は，表 2.4(b)の真理値表から明らかなように，負論理のAND回路は，正論理では**表 2.5(a)**に示すOR回路そのものとなっていることです。実際に使用するICも当



(a)　正論理のOR回路　　(b)　負論理のOR回路

図2.9　OR回路のシンボル

然，OR素子である7432を用いることになります。

すなわち，配線図上では負論理のAND回路が，実際の回路では**図2.9(a)**のように構成されています。したがって，実回路に合わせて図2.9（a）のように図面が書かれていたら，まったく何を意味する回路か他人にはわからなくなります。

この点を避けるため，設計図面を書くときの心がけとして，正しい負論理回路の記号を用いなければなりません。そのためにも負論理回路に早く慣れるようにして下さい。

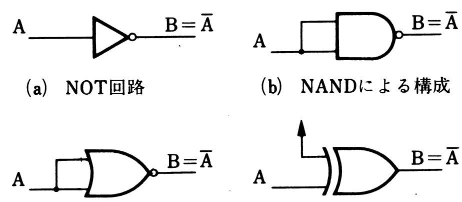
また，同様に負論理のOR回路は，表2.5(b)に示す構成となり，その回路記号は，図2.9(b)に示すように,負論理のORであることがわかる○印が入力出力端子につきます。

■NOT回路の構成

“HighをLow”，“LowをHigh”に変える回路をNOT回路とよび，**図2.10(a)**に示す三角印で示します。通常NOT回路は，TTL-IC 7406で実現しますが，その他図2.10（b）～(d）に示す7400(NAND回路)，7402(NOR回路)，7486(排他的論理和（EOR）回路）を用いて実現します。

すなわちNAND, NOR回路を，2入力を連結して1入力として使うことによりNOT回路を形成し，排他的論理和回路は**表2.6**に示す真理値表となります。したがって，片方の入力を“High”に設定しておくことによりNOT回路を実現します。

このような使い方は，2入力を1入力として使うので，無駄に感じられますが，1チップに4個あて実装されていますので，使っていない余りの回路を利用して，7404をほかに用意せずにNOT回路を実現することができます。しばしば利用される方法です。



(a)　NOT回路　(b)　NANDによる構成　(c)　NORによる構成　(d)　EORによる構成

図2.10　NOT回路の構成方法

表2.6　排他的論理和EORの真理値表

$C=A\oplus B$

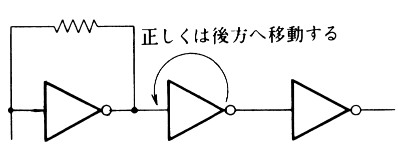
A	B	C
Low	Low	Low
High	Low	High
Low	High	High
High	High	Low

《正しい図面の書き方》

規格に合った正しい配線図を書くことは，大変重要な作業です。紙面の都合上，その詳しい書き方は割愛し，もっともやっかいな負論理信号を示す○印のつけ方について，その“こつ”を紹介します。

それは，**負論理信号は負論理信号同士が連結される**という原理に従い，原則として○印から出た信号は○印に入力されるということです。例えば，図2.6でも同じ動作だからといって，**付図2.4**のように書いてはいけません。

その他，さまざまな規則がありますので，正しい図面の書き方については，参考文献[2-4]を読んで下さい。



付図2.4　誤った図面の書き方

第1部第3章

マイクロ プロセッサ 3

ここでは，マイクロプロセッサに関する簡単な説明と，動作させせるために必要なマシン言語などについて述べます。

3.1 マイクロプロセッサについて

手のひらにのる小さなICが，マイクロプロセッサとよばれるコンピュータの頭脳部分を担当する素子です。このマイクロプロセッサこそマスコミによりさわがれているマイクロエレクトロニクス革命の担い手です。

■マイクロプロセッサ

マイクロプロセッサは，演算部（アキュムレータ）のビット数，取り扱うデータのビット数およびアドレスのビット数により，各種の仕様能力のものが市販されています。

すなわち，1ビット，4ビットの演算部構成のものは主として家電，工作機械などの制御用に利用され，マイコンブームを引き起こしたのが8ビットCPU構成のマイクロプロセッサです。

現在，16ビットCPUの入門記事や使い方に関する特集が盛んに紹介されており，これからの主流となるものと考えられます。しかし，手軽に利用できる状態になるのは数年後であり，しばらくは8ビット時代が続くものと考えられます。

8ビットCPUとしては，8080（インテル社），Z-80（ザイログ社），6800（モトローラ），6502（シナテック）が広く用いられています。この8ビットマイクロプロセッサに関しては，多くのマイコン雑誌やマイコン関係の書籍，例えば産報出版の電子科学シリーズにおいては，マイクロコンピュータの使い方（シリーズNo. 61），マイクロコンピュータ・ソフトウエアの設計（No. 82），マイクロコンピュータの開発技法（No. 65）がありますので，詳しい紹介は省略します。

■Z-80マイクロプロセッサ

ここで紹介する手作りコンピュータでは，Z-80を用いることにします。Z-80に関する書籍[2-1),2-2),2-3)]が多く出まわっていますので，ここでは簡単にZ-80の特徴について紹介します。

Z-80は80系に属し，8080の命令を含み，さらに機能が向上しています。そのため，PC 8001（NEC），MZ-80（シャープ），M 100（ソード社）などの多くのマイコンのCPUとして利用されています。

多くのマイコンに搭載されていることにより，各種のZ-80を用いたソフトウエアが供給され，価値の高いマイコンが作製できることになります。とくに，アセンブラが共通に使えることが強みとなっています。

また，機能としてはリフレッシュ端子があることで，ダイナミックRAMが簡単に利用できることになります。そのため，安価に市販されるようになった64 K DRAM（M 5 K 4164 NS-20）を8個利用して，64 Kバイトのメモリ容量

が得られることになります。Z-80の有利な点としては，次の点があげられます。

① 単一電源（5 V）であること（8080は他に12 Vと－5 Vが必要）
② CPU支援用の周辺ICが不必要（8085はアドレスとデータを分離するICが必要）
③ 割込みが強力であること
④ ブロック転送命令があること

その他の特徴としては，周辺ICのファミリが充実していることや，各種の命令サイクルが比較的高速である点などがあげられます。

■Z-80の仕様

Z-80のすべての入出力はTTLレベルであり，消費電力は最大200 mAとなっています。クロック周波数はZ-80の2.5 MHz仕様，4 MHzのZ-80 A，さらに6 MHzで動作するZ-80 Bが市販されています。本システムでは，4 MHzで動作させますのでZ-80 Aを用います。以下，本書ではZ-80 Aを略してZ-80と呼ぶことにします。

3.2 マイクロプロセッサの動作

第3章　マイクロプロセッサ

マイクロプロセッサの基本的な動作原理について簡単に述べます。マイクロプロセッサは，メモリ内のあるアドレスから命令コード（実体は8ビットの2進数）を読み込み，解読し，その命令を実行し，終了後，次の命令をふたたび読み込み，実行を繰り返します。

命令を読み込むサイクルを**フェッチサイクル**，また1つの命令の実行時間を**サイクルタイム**とよび，基本クロック周波数の何サイクル分かが**サイクル数**として与えられます。Z-80の命令は，最大4バイトの命令コード長から構成されています。

■16進表示

Z-80のアドレスはA_0からA_{15}の16ビット（2バイト）で与えられ，最大2^{16}の64 Kバイトとなっています。このアドレスの表示として，2進数で示すと16桁も必要で，扱いにくく，そのため16進数が用いられます。

16進数は，**図3.1**に示すように4ビットずつ区切り，各4ビットのビットパターンに応じて，**図3.2**に示す0からFの英数字1文字が対応します。0から9までは，通常の10進数と同じで

10進	16進
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

図3.2　16進数の表示

2				A				E				7			
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	1	0	1	0	1	0	1	1	1	0	0	1	1	1

図3.1　16進数の利用

すが，10はA，11はBが順次対応し，15にFが対応します。

例えば図3.1に示すビットパターンは，16進数で2AE7と書きます。また，10進と16進数を混同しないように，16進数であることを示すのに最後にHをつけて表示します。

つまり1234Hは，16進の1234であり，10進数の1234とはまったく異なった値をとることになります。この16進を用いると，Z-80のアドレス空間は，0000H番地からFFFFH番地の最大64Kバイトのアドレスとして表現できることになります。16進数表示は，アドレス空間のみならず，データの値や命令コードの表現などにも利用されますので，16進数表現に慣れるようにして下さい。

■機械語

マイクロコンピュータは，あるアドレス空間上のデータを命令語として読み取り，解釈し実行します。例えば1バイト命令であるBレジスタの内容をAレジスタに入れる命令は78H，またCレジスタの内容とAレジスタのORをとる命令は，B1Hとなっています。したがって0番地に78，1番地にB1を書き込んでおき，0番地から実行すればBレジスタの内容をAレジスタに入れ，その内容とCレジスタとのORを求める演算が行なえることになります。

このように，マイコンのメモリ上に指定した動作を行なう16進数2桁から最大8桁の命令語の数字を配置したとき，その16進数列を**機械語**（Machine Language）[*]またはオブジェクトとよんでいます。機械語は，マイクロプロセッサを直接動作させるためのもっとも基本的な言語です。この機械語は，手動のEPROMライタを用いて直接ROMに書き込む場合や，モニタを用いて簡単なプログラムを実行させるときに利用されます。

■アセンブリ言語

機械語は，ただの数値列ですので，直接プログラムを作る言語としては慣れるまで利用しにくいものです。数値列を見ても，どのような動作をするか直観的には理解できません。そこで，プログラム内容が比較的わかりやすく，また命令語と1対1に対応させた高級言語がアセンブリ言語とよばれるものです。

例えば，アセンブリ言語でBレジスタの内容をAレジスタに移す命令は

“78H”ではなく，“LD C,A”

と記述し，その命令内容と命令言語が類似しています。そのため，初心者にも覚えやすいので，できればアセンブリ言語を理解するようにして下さい。

■ニーモニック

アセンブリ言語で書かれたプログラムを機械語へ翻訳するのがアセンブラシステムです。またアセンブリ言語の命令の記述（例えば，LD A,C）をニーモニック（Mnemonic）と呼びます。

機械語は，Z-80 CPUならすべて共通の言語として利用できますが，アセンブリ言語は，そのシステムが異なると一部異なったニーモニックとなります。代表的なニーモニックとしてはザイログ系，80系，ソード系などがあります。

本書ではザイログアセンブリ言語を用います。どのアセンブリ言語でも，必ず対応するニーモニックコードがありますので，どれか1つのニーモニックコードを覚えれば，他のニーモニックコードも大した労力なく利用できるようになります。

---

* 機械語：正式にはコンピュータが直接理解できる命令語であり，0と1だけによる2進数で構成されたものをいいます。しかし，ここでは簡単のため，16進表示に変換したものも機械語とよぶことにします。

《RS232Cインターフェース基板の作り方》その1

ここで製作したマイコンをインテリジェント端末機として使いたい場合や，音響カプラを用いて電話回線でのデータの伝送を行なおうとする場合は，RS 232 Cインターフェースが必要になります。RS 232 Cはモデムとデータ端末装置間のディジタル2進直列データ，制御信号，タイミング信号の転送を行なうためのインターフェースの規格です。

RS 232 Cは，相互接続時に信号電圧やインピーダンスなどについての細かい規定があり，JISにも同様な規格（JIS―6361“モデムと通信制御装置およびデータ端末装置とのインターフェース”）があります。しかし，ここではTTLレベルからRS 232 Cへの変換用の専用ICを用いることにします。

RS 232 Cの信号線は300ビット／秒の音響カプラを用いて，**付表A**に示した信号仕様とします。また，RS 232 Cでは接続コネクタも規定されています。コネクタの規格を**付図A**に示します。表のピン番号はこの図のコネクタ番号に対応しています。

付録10に手作りコンピュータにおけるRS 232 Cインターフェース基板の全回路図を示します。**付図B**に示した8251 AはUSART（Universal Synchronus／Asynchronous Receiver／Transmitter）と呼ばれるICで，CPUから並列データを受けとり，直列に変換して送出したり，逆に直列データを並列に変換して，CPUに送出します。またUSARTの状態を，モデムや端末に知らせるための信号の制御を行ないます。

SN 75188とSN 75189は，TTLコンパチブルであるUSARTの入出力電圧レベルを，RS232Cの電圧レベルに変換しています。直列信号の通信速度は，8251 A（ピン配置は付図B参照）のT_xC，R_xC端子に加えられるクロックにより決定されます。このクロックを作るのがボーレートジェネレータCOM 8416です。　（59ページに続く）

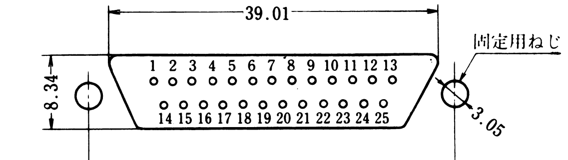
付表A　300ボーのモデムとの信号

（a）

ピン番号	信号名称	略号	信号方向	動作
1	保安用接地	FG	端末モデム	装置の筐体と接続し，外部地気に接続
2	送信データ	SD	→	端末が送信しようとするデータ
3	受信データ	RD	←	端末が受信しようとするデータ
4	送信要求	RS	→	DRがONであればモデムは送信状態になる
5	送信可	CS	←	モデムが送信状態にあるかどうかを示す　ON＝送信可　OFF＝不可
6	データセットレディ	DR	←	ONでモデムが送受信可能
7	信号用接地	SG		FG以外のすべての相互接続信号の基準

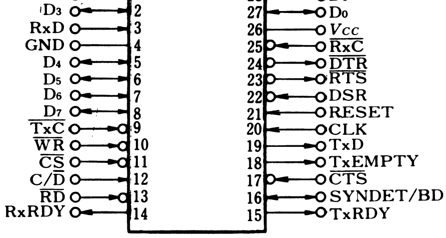
（b）

電圧V_1	データ	タイミングおよび制御信号
＋3V以上	0	ON
＋3V以下	1	OFF



モデム側のコネクタ（規格名＝JAE DB-2CSA）はピン形状がめす形で，接続ケーブル側はおす形とします

付図A　使用するコネクタの形状，外形寸法およびピン配列



付図B　8251 Aのピン配置

第1部第4章

基本回路の作り方　4

ここでは，フロッピーディスクコントローラ部分を除いて，マイコンとして動作する主要部分の作製手順について述べます。この部分が完動すれば，手作りコンピュータとしての主要部分は完成します。この基本回路の部分は，マイコンの基本的な働きを理解するためにも重要です。

第4章　基本回路の作り方

4.1 CPU基板の作り方

手作りコンピュータの頭脳部分に相当するのがCPU基板です。このCPU基板は，巻末の折込みページ**図4.1**に示す配線図から構成されています。主なICはZ-80マイクロプロセッサ，2Kバイトの EPROM (2716) で，メモリ (EPROM またはRAM) から命令を読み込み，解釈し，実行する作業を行なっています。この基板の動作および各素子の役割について説明します。

4.1.1　Z-80 CPUの使い方

CPUはZ-80で，40ピンのパッケージに実装されています。**図4.2**にピン番号を示します。

A_0～A_{15}はアドレスバスで，A_0が最下位ビット，またA_{15}が最上位ビットとなっており，0から65535 (64 K) 番地までのメモリをアクセスすることができます。

またD_0からD_7は，1バイトのデータバスを構成しており，トライステート*出力をもつ双方向性で，CPUと主記憶あるいは外部入出力装置との間で，データのやりとりを行なう端子です。

$\overline{\mathrm{BUSRQ}}$ (BUS ReQuest) 信号と$\overline{\mathrm{BUSAK}}$ (BUS AcKnowlege) 信号は，アドレスバスとデータバスを制御する信号です。$\overline{\mathrm{BUSRQ}}$は，CPUがバスを他のシステムに明け渡すときの外部機器からCPUへの要求信号です。

また，$\overline{\mathrm{BUSAK}}$はCPU側が$\overline{\mathrm{BUSRQ}}$に応答してバスを明け渡したことを外部に知らせる信号です。

例えば，DMA (Direct Memory Access) 伝送においては，CPUを介さず直接主記憶と外部入出力装置との間でデータの転送を行ないます。その際，CPU側が出すデータやアドレスとの競合を避けるために，CPU側のバスを切り離す操作を行なう場合などに利用されます。

図4.1において，$\overline{\mathrm{BUSAK}}$信号が図面内**7B**にある3つのトライステートバッファ (74 LS 367) に加えられ，これによってアドレスバスを制御し，**4D**にある74 LS 245に接続することによってデータバスを制御しています。

$\overline{\mathrm{RFSH}}$ (ReFleSH) 信号は主記憶にダイナミックRAMを用いるときにリフレッシュ信号と

* トライステートは，"High"，"Low"および入力端子に負担をかけないハイインピーダンス状態のどれか1つの状態を選択できる出力構成をいいます。

access 出入り

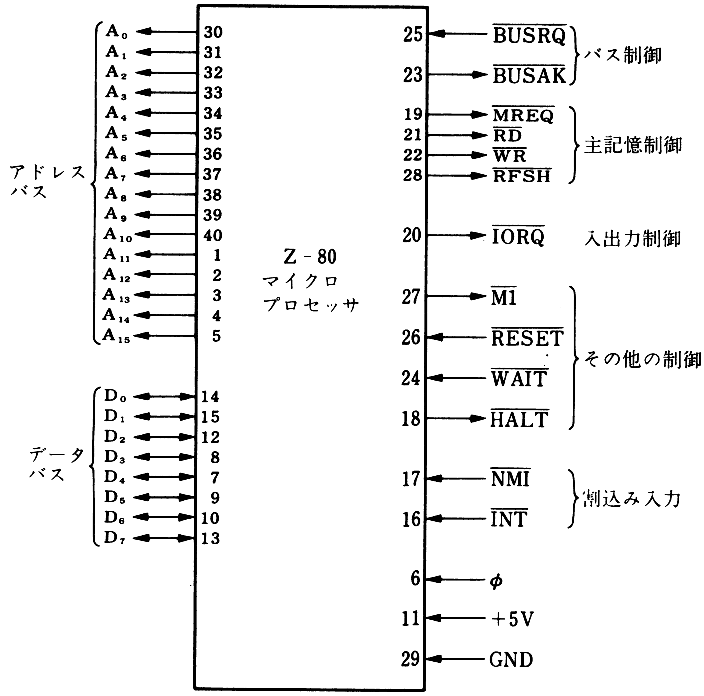


図4.2　Z-80のピン配置

して使います。ダイナミックRAMは，容量への蓄積電荷を利用した記憶素子なので，内容が消えない時間内(約2ms以内)に再書込み(リフレッシュ)をする必要があります。

ここで作製するシステムでは，64Kのダイナミック RAM を用いていますので，リフレッシュ信号を利用します。

リフレッシュ信号は，2msを越えないように，$\overline{\text{RFSH}}$信号が現われ，かつ$\overline{\text{MREQ}}$信号が動作状態になれば，アドレスバスの下位7ビット(CAS信号)を用いて1リフレッシュサイクルを実行します。すなわち，図面内**3F**のフリップフロップ(74 LS 74)回路が利用されています。

$\overline{\text{RFSH}}$信号は，すべてのフェッチサイクル(命令読込み)ごとに動作し，リフレッシュレジスタ(Rレジスタ)の内容が1つずつ増加し，新しいリフレッシュ番地が与えられます。また$\overline{\text{M1}}$信号は，Z-80が命令の読出しサイクル(フェッチ)にあることを示す信号です。

$\overline{\text{INT}}$信号と$\overline{\text{NMI}}$信号は，割込み(INTerrupt)に関する信号で，$\overline{\text{INT}}$は，通常の割込みを要求する外部入出力装置から与えられ，$\overline{\text{NMI}}$信号は，マスク不能(Non Maskable)要求です。

$\overline{\text{NMI}}$信号が入ると，プログラムカウンタの内容を，スタック内へ待避し，CPUは制御を0066H番地へ移し，命令の実行が66Hからはじまることになります。

$\overline{\text{RESET}}$信号は，CPUのリセット用の信号で，CPUの動作はプログラムカウンタPCが0となり，0番地のプログラムから再実行されます。

$\overline{\text{WAIT}}$信号は，低速の記憶素子または入出力装置を利用するときに用いられ，$\overline{\text{WAIT}}$信号が“Low”となっている場合は，CPUは次の命令に進まず，$\overline{\text{WAIT}}$信号が“High”になるのを待ちます。ここでは，CPUが4MHzクロックで，低速の記憶素子であるEPROM 2716(450 ns)を使用しますので，M1サイクルでCPU側に$\overline{\text{WAIT}}$信号を出します。

このＭ１サイクルで$\overline{\text{WAIT}}$信号を発生させる部分が図面**1D**にある２段のフリップフロップ回路です。また，低速の入出力装置であるフロッピーディスクを用いますので，フロッピーディスクとCPU間のデータのやりとりを行なう際の時間調整をするのに$\overline{\text{WAIT}}$信号をかけます。このフロッピーディスク制御基板からの$\overline{\text{WAIT}}$信号が外部端子B 21（図面**1C**）から与えられる$\overline{\text{WAIT}}$信号です。

■**入出力動作信号**

CPUの命令実行動作に伴う信号が$\overline{\text{RD}}$（ReaD），$\overline{\text{WR}}$（WRite），$\overline{\text{MREQ}}$（Memory

《４MHz動作とROM》

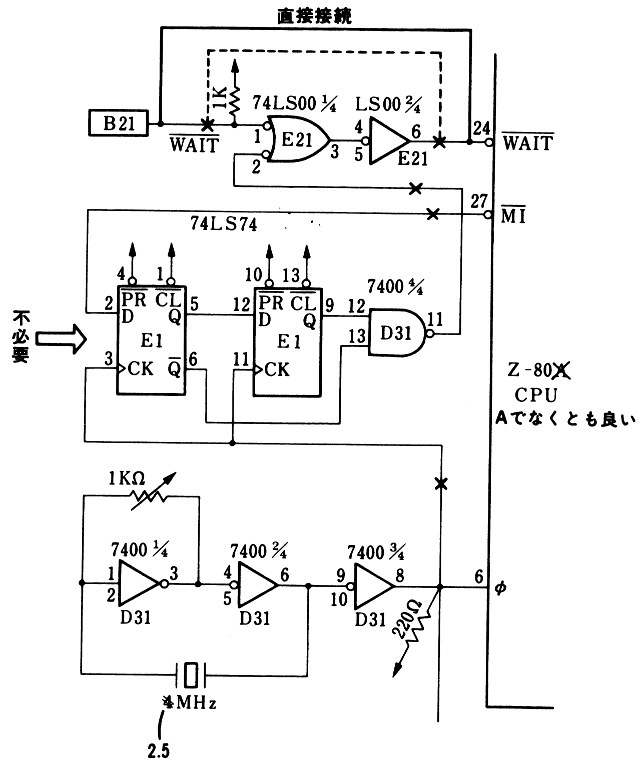
Z-80 Aを用いて４MHzで動作させる場合，最短のアクセスタイムは355 ns以下でなければなりません。この値は，使用するEPROM 2716（450 ns）では追従できないことになります。したがって，この値を満足させるためには高価な2716-1（インテル）または，MCM 27 A 16（モトローラ）などの350 nsのアクセスタイムのものか，あるいは，４Kバイトになってしまいますが，2732 A (250 ns) を用いなければなりません。

そこで，ここでは，メモリアクセス命令であるＭ１サイクルでWAITをかけて，アクセスタイムを補っています。

《2.5 MHzクロック》

図4.1の配線図では，４MHzで動作する仕様となっています。この仕様でも，許容値に入っていますので，確実に動作はします。しかし，**マイクロプロセッサは余裕をもって動作させる**のが好ましい状態です。そのため，初心者は，水晶を入れ換えるだけですみますので，2.5 MHzで動作させることをおすすめします。

2.5 MHzで動作させた場合，当然のことながら図面中**1D**にある74 LS 74から構成されるＭ１サイクルのWAIT信号は省略できることになります。そのため，CPUの$\overline{\text{WAIT}}$入力には**付図4.1**のように直接フロッピーディスクコントローラからの$\overline{\text{WAIT}}$信号（ソケット端子B 21）のみが入力されます。



付図4.1　4 MHz仕様から2.5MHz仕様への変更

REQuest)，$\overline{\text{IORQ}}$（IO ReQuest）の4つの信号です。

$\overline{\text{RD}}$，$\overline{\text{WR}}$は，トライステート端子で，CPUの動作が読出し動作であるか，また書込み動作であるかを示す信号です。

$\overline{\text{MREQ}}$信号は，主記憶データ要求信号であり，$\overline{\text{MREQ}}$信号が“Low”および$\overline{\text{RD}}$が“Low”であれば，主記憶からの読出しが行なわれます。この信号が図4.1の図面中**3D**にある負論理AND回路（D 21）で作られる$\overline{\text{MEMR}}$（MEMory Read）信号です。

同様にして，主記憶への書込み動作は，負論理AND回路（D 21）で作られる$\overline{\text{MEMW}}$（MEMory Write）信号です。

一方，入出力装置に対する入出力命令を発生する信号が$\overline{\text{IORQ}}$です。この$\overline{\text{IORQ}}$信号は，負論理のトライステート端子で，$\overline{\text{IORQ}}$信号が“Low”となったとき，入出力アドレス上の下位8ビット（A_0～A_7）上に各入出力機器に対して設定した番地が出力され，機器選択が行なわれます。

これより，$\overline{\text{IORQ}}$が“Low”かつ$\overline{\text{RD}}$が“Low”のとき，入出力機器からの読出し信号$\overline{\text{IOR}}$（I／O Read）が得られます（同図面位置**3E**）。同様にして，$\overline{\text{IOW}}$（I／O Write）信号が得られることになります。

■クロック発生部

図4.1の図面上**1E**にある3つのNOT回路および4 MHzの水晶と1 kΩのボリュームにより4 MHzのクロックϕを発生しています。1 kΩのボリュームを調整して，4 MHzの矩形波を発振させます。その設定方法は，**図4.3**に示すようにデューティが0.5，すなわち“High”状態の幅と“Low”状態の幅が極力同一となるように設定します。

この際，注意すべき点はクロックの電気的特性です。他のZ-80の入出力端子はTTLコンパチブルですが，図4.3に示すように，クロックϕの“High”レベルは4.4 V以上，また“Low”レベルは0.45 V以下と定められています。そのため，7400を使用した場合，“High”電圧の不足を招くことがあります。

そこで，Z-80のクロック入力部を220 Ωの抵抗で電源にプルアップしておきます。また，クロック信号の波形条件としては，“High”レベルがZ-80では180 ns以上，Z-80 Aでは110 ns以上のパルス幅がなければなりません。

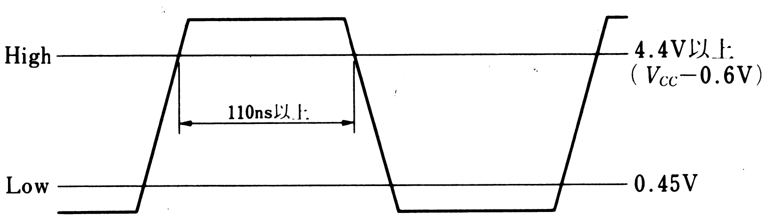


図4.3　クロックϕの電気的条件

4.1.2　EPROMの使い方

2 Kバイト（16 Kビット）の紫外線で消去可能なUVEPROM（Ultra-Violet Erasable Programmable Read Only Memory，略してEPROM）を用いて，電源開始時およびリセット時の再スタート用プログラムを書き込んでおき，動作させます。

プログラムは，2 Kバイトのメモリ容量なので，アドレスが0000番地から07 FF番地の範囲で使用されます。

このEPROMのアドレス設定が図4.1上の**3C**にある74 LS 42であり，A_{12}～A_{15}の4ビットを“Low”とし，74 LS 32（同図面内**3C**）でA_{11}も“Low”であるときEPROMが動作するようにしています。

また，**4C**にあるAND回路（D 1）の一方の入力信号$\overline{\text{BANK SELECT}}$は，EPROMのプログラムをRAM領域に移行後，EPROMを切り離

《TTLコンパチブル》

論理HighとLowに関して，直流特性条件が，

High状態　2.0 V以上

Low状態　0.8 V以下

と定められ，互いに入出力可能な端子をＴＴＬコンパチブルといいます。

《74 LSシリーズの使用》

TTL-ICにはノーマル型の74シリーズのほかに，74 LS，74 Sなどがあります。このLSは，Low power Shotkey clampedの略で，ノーマル型と変わらない速度をもち，かつ低消費電力化されたものです。

通常ノーマル型とLS型が多く用いられます。ノーマルとLSの差は，負荷駆動能力にあります。ノーマル型は，Lowレベルの際の負荷電流I_{IL}は−1.6 mA程度ですが，LS型シリーズの特性は，

I_{IH} (max) = 20 μA （V_{IH} = 2.7 V）

I_{IL} (max) = −0.4 mA （V_{IL} = 0.4）

が標準の入力電流です。

この差が，CPUや各種インターフェースLSI側の駆動能力に影響を与えることになります。例えば，図4.1に示したZ-80のアドレスバスには，負荷として74 LS 42,2716および74 LS 367が接続されています。そして，Z-80のアドレスバスの負荷駆動能力は，1.8 mAであるため，

I_{OL} = 1.8 mA − 04 mA − 0.4 mA − 0

↑ CPUの能力　↑ LS42　↑ LS 367　↑ 2716

I_{OL} = 1.0 mA

の余裕があります。これに対し，誤ってノーマル型である7442および74367を使用した場合は，

I_{OL} = 1.8 mA − 1.6 mA − 1.6 mA

↑ CPUの能力　↑ 7442　↑ 74367

= **−1.4** mA

となり，完全にオーバーロードとなります。TTLを接続させるときは，この負荷駆動能力に関する設計が重要となります。各ICの入出力特性は，TTL-ICの規格表に掲載されていますので，自分で設計する際には必ず負荷状態を確認して下さい。

いずれにしても，CPU，PPIなどのMOS-ICと接続する場合には，とくに指定がない限り**SN 74 LS型を利用すること**が肝要です。

付表4.1に使用ICとの入出力直流特性を示します。

付表4.1　主な素子の電気的直流特性

素子名	出力電流特性		入力電流特性	
	I_{OL}(mA)	$-I_{OH}$(μA)	$-I_{IL}$(mA)	I_{IH}(μA)
74LS××	8	400	0.4	20
74××	16	400	1.6	40
2716	2.1	400	0	10
2114	2.1	1.0 mA	0.01*	10

* データバスのみ

し，全領域（64 Kバイト）を読み書き可能なRAM領域とするために使用するバンク切換え信号です。

すなわち，$\overline{\text{BANK　SELECT}}$信号が“Low”となると，2716のチップセレクト信号（$\overline{\text{CS}}$）が“High”となり，ROMが選択されなくなり，切り離されます。そのため，PROMが占めていたアドレス0000番地から07 FF番地も，RAM領域が使えることになります。このRAM領域に切り換える信号が，**4C**にある7400（D 1）の6番ピンから出るROM　SELECT信号（端子番号A17）です。

■ **2716の動作**

2 KバイトのEPROM 2716のピン配置を**図4.4**に示します。2716は5 V単一電源で動作し，入出力端子すべてがTTLコンパチブルのトライステート出力となっています。メモリの読出し動作は**図4.5**に示すようにPD/PGM（Power Down／ProGraM）が“Low”でかつチップセレクト信号$\overline{\text{CS}}$が“Low”となったとき，A_0からA_{10}で指定されたアドレスの内容が出力されることになります。

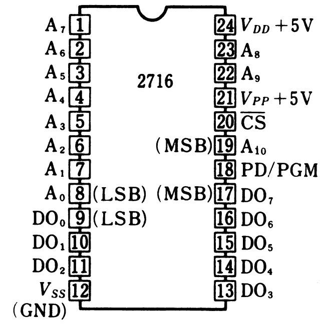


図4.4　2716のピン配置

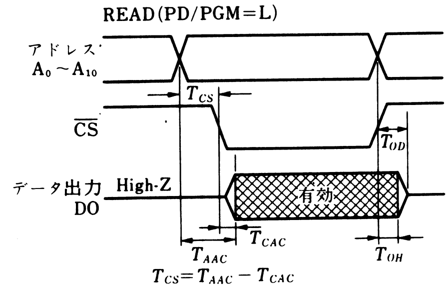


図4.5　2716のREAD動作波形

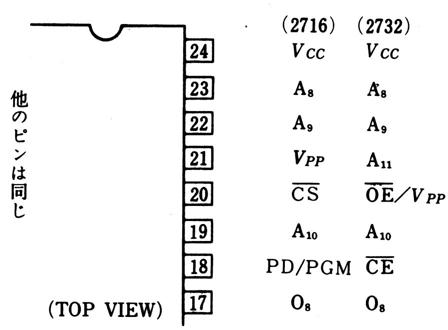
《2716と2732》

2716は2 Kバイト，2732は4 KバイトのEPROMです。手作りコンピュータシステムでは2716を使用するように設計してありますが，最近2732が安価に出まわり，EPROMの主流になりつつあります。

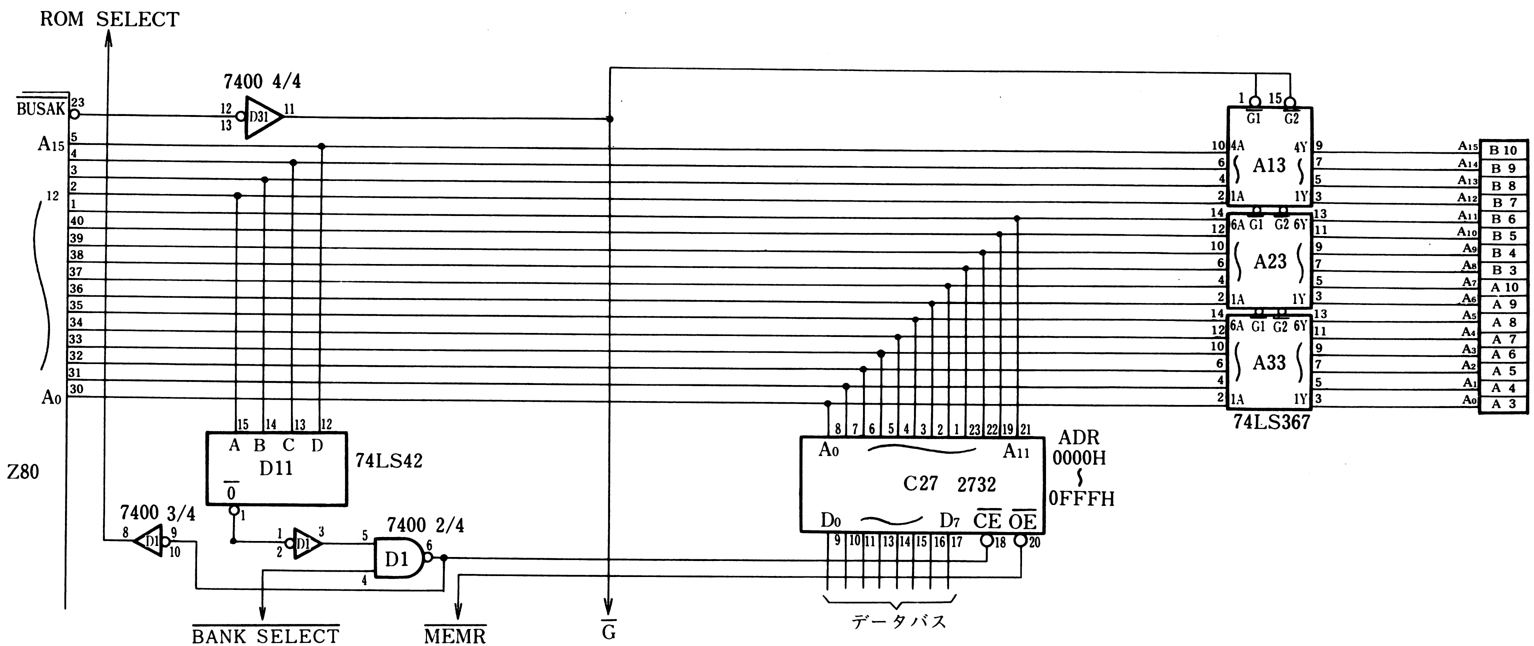
この2732と2716との差は，**付図4.2**に示すように⑱～㉑ピンの役割が変わっているだけです。[4-5)]

メモリの読出し動作では，2732と2716は$\overline{\text{OE}}$と$\overline{\text{CS}}$，$\overline{\text{CE}}$とPD／PGMが同じです。したがって，21番ピンに対して**2716が5 Vだったのを，2732はA11（アドレス11）とするだけで簡単に変更できることになります。**

2732を使用する場合には，図4.1の図面の一部を**付図4.3**に示すように変更して下さい。



付図4.2　2716と2732のピン配置の相違



付図4.3　4 K EPROM 2732の利用

4.2 キーボード制御基板の作り方

キーボードとCPUのインターフェースとして図 **4.6** に示すキーボード基板を製作します。この回路では，並列入出力用のインターフェースIC 3255 Aを使用します。
8255A

4.2.1　8255 Aの使い方

8255 Aは，8ビットマイクロコンピュータシステムのために設計された汎用プログラマブル入出力素子で，PPI(Programmable Peripheral Interface) とよばれています。8255 Aのピン配置を図 **4.7** に，またブロック図を図 **4.8** に示します。

この素子は，図4.8におけるPA_0からPA_7，PB_0からPB_7およびPC_0からPC_7までの合計8×3ビットの並列入出力動作が行なえます。この24ビットは，8ビットずつ3つのグループまたは12ビットずつの2つのグループで独立にプログラムされ，3つの主なモードで動作することができます。

すなわち，第1のモード(モード0)は次に述べるように，各グループの12のピンが入力または出力として4種類にプログラムされます。ここで作製するコンピュータでは，モード0を利用します。

モード1は割込み信号の制御用として利用する方法であり，モード2は両方向性バスモードです。

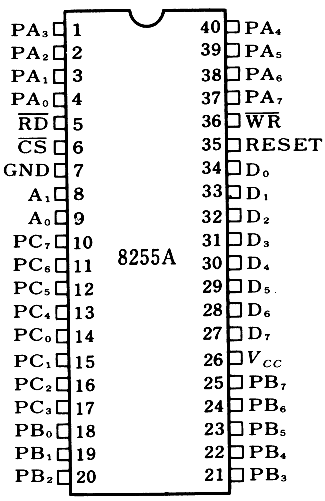


図4.7　8255 Aのピン配置

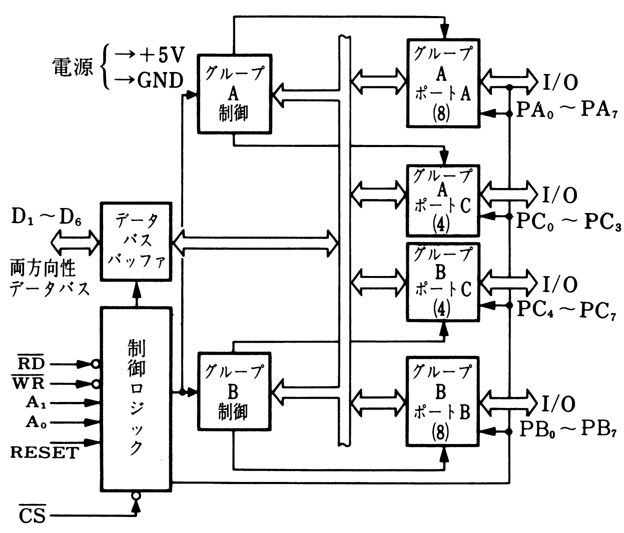


図4.8　8255 Aのブロック図

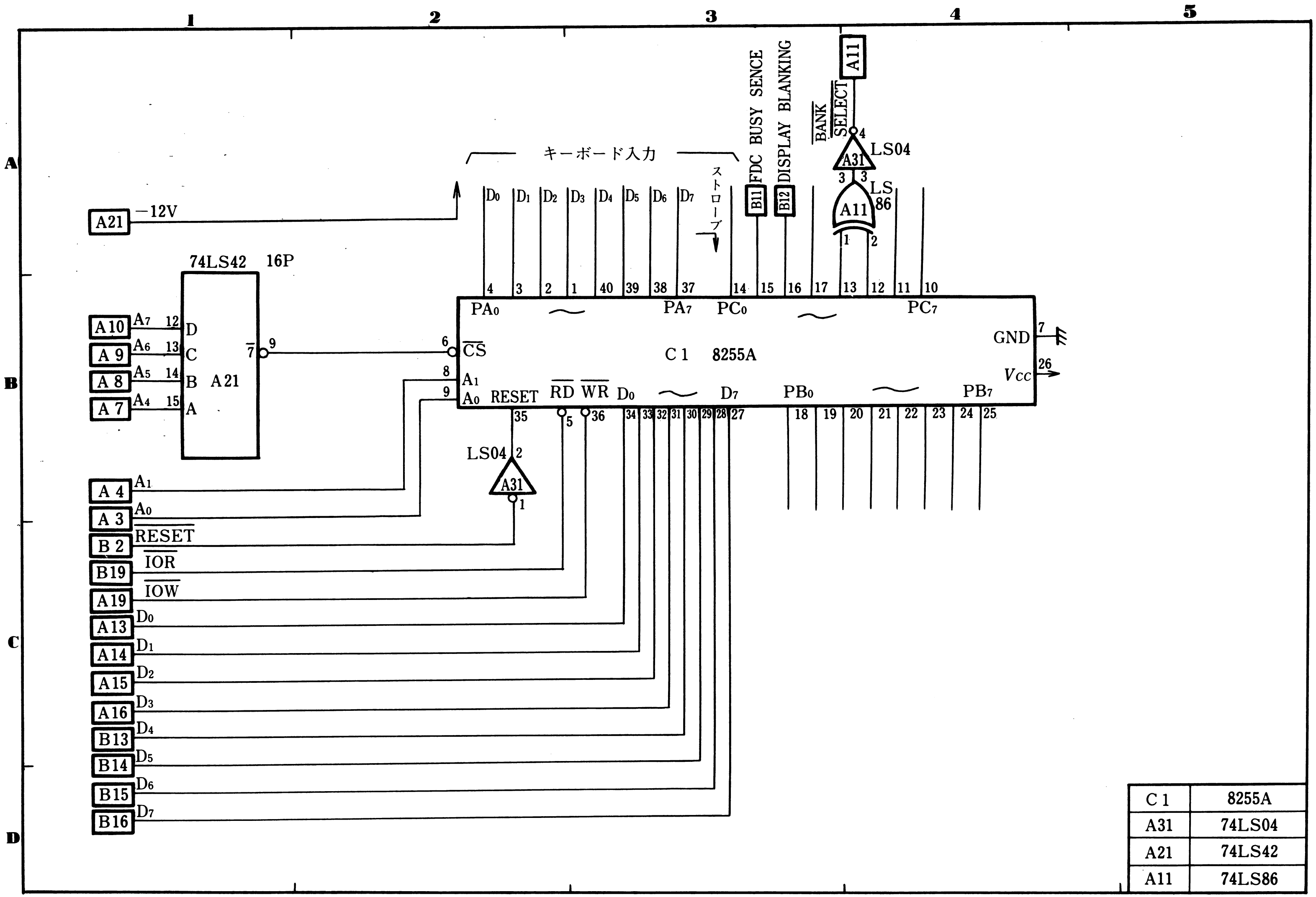


図4.6　キーボード制御用基板の配線図

■ 8255 Aの動作

8255 Aは，5 V単一電源であり，完全なTTLコンパチブル端子となっています。また出力のソース電流が1.5 Vで，1 mAの能力があり，これはプリンタやディスプレイなどのダーリントントランジスタを直接駆動する能力をもっています。

データバスD_0からD_7は，トライステートバッファから構成され，データはCPUの入出力命令実行によって，このバッファを経由して送信／受信されます。また，8255 A制御用のデータ(コントロールワード)やステータス情報もここを経由して転送されます。

RESET端子は，8255 Aをリセットする信号で正論理なので，“High”が入ることによりリセットがかかり，24本の入出力端子は，入力ポートとして動作します。このリセット端子は，開放（何も配線しない状態）では“High”となることがあり，リセットがかかる状態になることがあります。

したがって，動作しなくなる恐れがあるので，必ず使用時には“Low”となるような配線を必要とします。図4.6の配線図中**2B**にあるNOT素子(A 31)は，CPUのリセット信号が負論理$\overline{\mathrm{RESET}}$なので，反転して8255 AのRESET信号を得るためのものです。

■ 8255 Aの選択

$\overline{\mathrm{CS}}$（Chip　Select）信号端子への“Low”入力で8255 AとCPU間の通信が可能となります。CPU基板のI/Oリクエストの際の下位アドレスA_0からA_7により，機器指定が行なわれます。

この指定を行なうのが図4.6上**1B**にあるTTL 74 LS 42で，A_7，A_6，A_5，A_4が“0111”になるとき，$\overline{\mathrm{CS}}$が“Low”となって，キーボード制御用8255 Aが選択されます。

$\overline{\mathrm{RD}}$は8255 Aからの読出し信号であり，この端子への“Low”入力で，8255 AがCPUのデータバスへデータやステータス情報を送出します。$\overline{\mathrm{RD}}$は，CPU基板の$\overline{\mathrm{IOR}}$信号が用いられます。

同様にして，$\overline{\mathrm{WR}}$は8255 Aに対する書込み信号であり，この端子へ“Low”を入力することでデータやコントロールワードが書き込まれます。$\overline{\mathrm{WR}}$には，$\overline{\mathrm{IOW}}$信号が印加されます。

他の制御入力を接続するときは，74 LS 42の7以外の出力端子を用いることにより，機器指定が行なえます。すなわち，74 LS 42の利用により，上位アドレスA_7，A_6，A_5，A_4を変えることで最大15台までの入出力機器が競合せずに選択できることになります（入出力機器の割当てを参照)。

■ポートアドレス

A_0，A_1は$\overline{\mathrm{RD}}$と$\overline{\mathrm{WR}}$との組み合わせにより，3

《入出力機器の割当て》

A_0～A_7の下位8ビットのアドレスバスを利用して，I／O命令における入出力機器指定を行ないます。8ビットを利用しますので，最大256台の機器を競合せずに選択可能ですが，その場合は8ビットのフルデコードが必要となります。

したがって，ここでは**付図4.4**に示すように下位4ビット中のA_0，A_1は，各機器の制御信号用，またA_2は，フロッピーディスク装置，A_3はCRTコントローラに割り当て，残りのA_4～A_7の4ビットをデコードして15台の機器を選択させることにします。

上位4ビット中，付図4.4に示すように“0111”をキーボードに，また“0000”をPROMライタに割り当て，残りのアドレスは読者がドットディスプレイや漢字ROMなどの他の装置を付属できるようにあけておきます。

(MSB) A_7 A_6 A_5 A_4	A_3	A_2	(LSB) A_1 A_0	割当て装置名
0 1 1 1	1	1	制御信号	キーボード
0 0 0 0	1	1		PROMライタ
1 1 1 1	1	0		フロッピーディスク
1 1 1 1	0	1		CRT制御
4ビットのデコード 最大15台まで可能	CRT専用	FD専用		

“Low”状態を0，“High”状態を1と略記

付図4.4　I／O命令におけるA_0～A_1アドレス配分

つのポートやコントロールワードレジスタを選択制御します。このA_0，A_1は，CPUアドレスの下位2ビット（A_0，A_1）に接続し，I/Oリクエストの際の設定アドレスにより選択されます。

8255 Aの基本動作を**表 4.1** に示します。すなわち，$\overline{RD}$が"Low"のときは 8255 AからCPUへの読込み動作であり，A_1，A_0がともに"00"でポートAのPA_0からPA_7端子の信号がデータバス上へ出力され，CPUに入力されます。さらに，A_1，A_0が"01"でポートBからデータバスへ，またA_1，A_0が"10"でポートCからデータバスへデータが出力されることになります。

同様にして，$\overline{WR}$が"Low"のときがCPUから 8255 Aへの書込み動作であり，A_1，A_0により 8255 Aのポートが選択されます。またA_1，A_0がともに"High"の場合は，後で述べるコントロールワードの設定に利用されます。

■ポートA，B，C

8255 Aは，A，B，C 3つのの8ビットポートをもっていて，コントロールワードの設定により，おのおの次のような汎用性をもつインターフェース動作が行なえます。

ポートA：1つの8ビットデータ出力ラッチバッファと1つの8ビットデータ入力ラッチ

ポートB：1つの8ビットデータ入力出力ラッチ/バッファと1つの8ビット入力バッファ

ポートC：1つの8ビットデータ出力ラッチ/バッファと1つの8ビットデータ入力バッファ。このポートCは，モードコントロールによって2つの4ビットポートに分割できます。各4ビットポートは，4ビットのラッチであり，これはポートAやBと組み合わせて，制御信号の出力あるいはステータス情報の入力として用いることができます。

■モード0の動作

モード0は，基本入力および出力動作であり，**図 4.9** に示すように，A，B 2つの8ビットポート，Cポートの下位PC_3～PC_0と上位PC_7～PC_4の2つの4ビットポートを設定することができ，いずれのポートも入出力どれにでも使用可能です。したがって，これらの4つのポートの組み合わせは16通りになります。

表4.1 8255 Aの基本動作表

A_1	A_0	$\overline{RD}$(IN命令)	$\overline{WR}$(OUT命令)	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A ➡DATA BUS
0	1	0	1	0	PORT B ➡DATA BUS
1	0	0	1	0	PORT C ➡DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS ➡PORT A
0	1	1	0	0	DATA BUS ➡PORT B
1	0	1	0	0	DATA BUS ➡PORT C
1	1	1	0	0	DATA BUS ➡CONTROL
					DISABLE➡FUNCTION
×	×	×	×	1	DATA BUS ➡3-STATE
1	1	0	1	0	ILLEGAL➡CONDITION
×	×	1	1	0	DATA BUS ➡3-STATE

各ポートの指定およびコントロールワードの設定は，**図 4.10** のように，A_1，A_0を"11"として，$\overline{WR}$により 8255 AへデータD_0からD_7を書き込むことにより行なわれます。すなわち，$D_7=1$でモード設定が行なわれ，D_5，$D_6=00$でグループAのモード0設定，そしてD_4，D_3でグループA中のポートAおよびポートCの下位4ビットの入出力設定が行なわれます。

同様にして，D_2，D_1，D_0を用いてグループB中のポートBおよびポートCの上位4ビットの入出力設定が行なわれます。

例えば，コントロールデータワードを**図 4.11** (a)に示すように，"10000000"と設定した場合は，すべてのポートが出力ポートとなり，同図(b)のように"10010001"とすれば，AとC（下位）が入力ポート，BとC（上位）が出力ポートとして

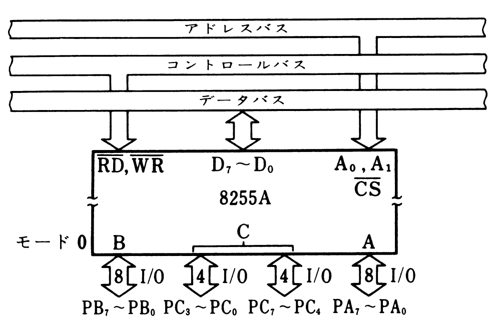


図4.9 モード0の動作

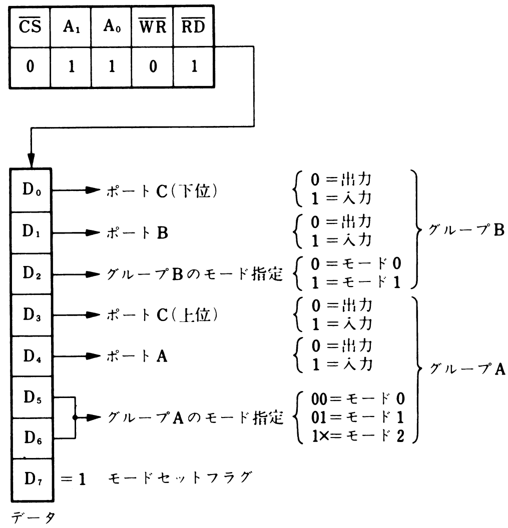


図4.10　コントロールワードの設定

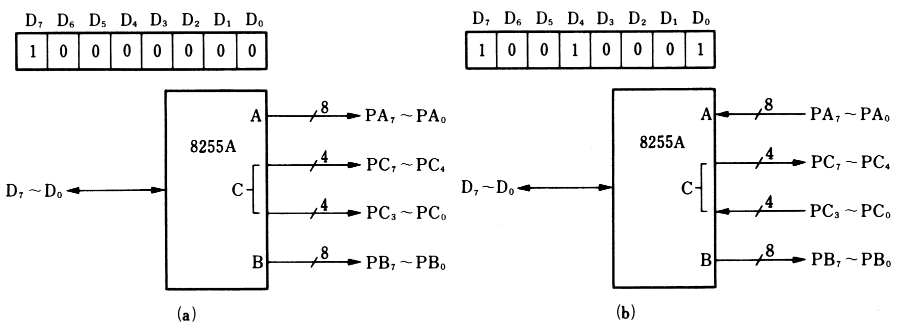


図4.11　コントロールワード設定の例

動作することになります。

4.2.2　キーボードの制御

キーボードからのデータの入力は，図4.6に示したように，ポートAとCを用いて行ないます。すなわち，ポートAの8ビットをキーボードからのアスキーコード入力に用い，ポートC(下位）中の1ビットPC_0をストローブ入力信号として用います。

ストローブ信号は，キーが押されたことを知らせる信号であり，キーが押されたことで“High”から“Low”へ変化します（機種により“Low”から“High”へ変化する場合や，パルス形状のものもあるので，購入したキーボードのデータシートを参照のこと)。ここでは**図4.12**に示すように，その立ち下がりを検出して，ポートAのコードをCPU側に読み取っています。本システムでは，ポートBはとりあえず使用しません。

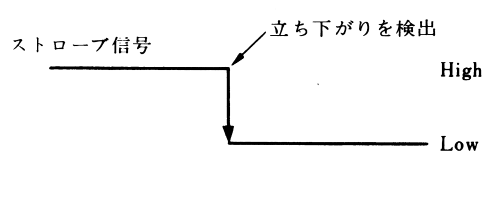


図4.12　ストローブ信号の検出

第4章 基本回路の作り方

4.3 CRT制御基板の作り方

手作りコンピュータの顔に相当するCRTディスプレイをコントロールするのがCRT制御基板です。CRT制御基板は，巻末折込みの**図4.13**に示す配線図のようになっており，CRT制御専用ICとして日立製のHD 46505を利用しています。

また，HD 46505は256×256のドットディスプレイのコントローラとしても利用できますが，ここで考えている手作りコンピュータでは，キャラクタのディスプレイ用として利用することにします。まずはじめに，ラスタスキャンディスプレイ方式について述べ，製作する基板の動作および使用するICの機能について説明します。

4.3.1 ラスタスキャン方式について[4-2)]

CRT制御基板の動作説明の前に，簡単にラスタスキャンディスプレイについて説明します。

■ラスタスキャン方式の原理

ラスタスキャン方式のディスプレイは，テレビ受像機と同様に，CRT（Cathod Ray Tube）の蛍光面上に電子ビームを当て，発光させるわけですが，このとき電子ビームをラスタスキャンとよばれる走査に従って偏向させることにより，画像を表示させるものです。

1つの画面は，**図4.14**のように左から右へ走査する水平走査と，上から下へ走査する垂直走査から構成されます。図中の実線の部分を表示期間，破線の部分を帰線期間とよびます。

図4.14に示すスキャン方式とは別に，現在放送用テレビ受像機などで使用されている走査方式にインタレーススキャン方式（飛越し走査）とよばれる方式があります。これは，**図4.15**のように前回走査した走査線の中間に位置する走査を行なう方式です。そのため，2回の垂直走査（奇数フィールド，偶数フィールド）で1画面（フレーム）を構成します。

同じ水平同期周波数を用いた場合，インタレース方式はノンインタレースに比べて，垂直方向の解像度が2倍に高くなりますが，飛越し走査のため，垂直走査の周波数が低いとちらついて見え，画質の低下を招くという欠点があります。

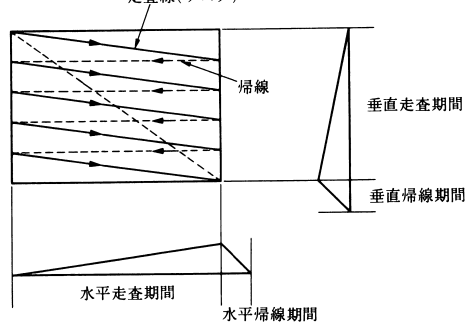


図4.14　ノンインタレース走査のラスタスキャン方式

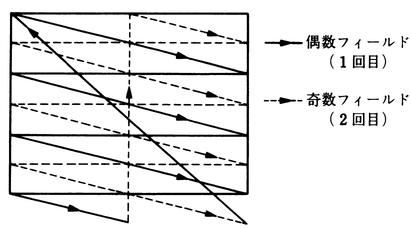


図4.15　インタレース走査のラスタスキャン方式

■キャラクタディスプレイの原理

ラスタスキャンディスプレイを用いて，キャラクタを表示する原理について述べます。キャラクタディスプレイのハードウエア構成を**図4.16**に示します。

CRTディスプレイ画面上に文字を表示しておくためには，CRT画面をたえず走査し，毎秒30～60画面を表示し続ける，すなわちリフレッシュする必要があります。そのため，表示する文字情報データをメモリに記憶しておき，この

情報に基づき，画面をリフレッシュする方式が用いられます。このような画像表示のためにデータを記憶するメモリをリフレッシュメモリとよび，通常RAMが用いられ，とくにVideo RAM(略してVRAM)とよばれます。

画面の表示は，CPUからVRAMに文字に対応するコードを書き込むことで行なわれます。また，VRAMに書き込むデータは，通常アスキーコードが用いられているため，そのままでは画面に文字として表示することはできませ

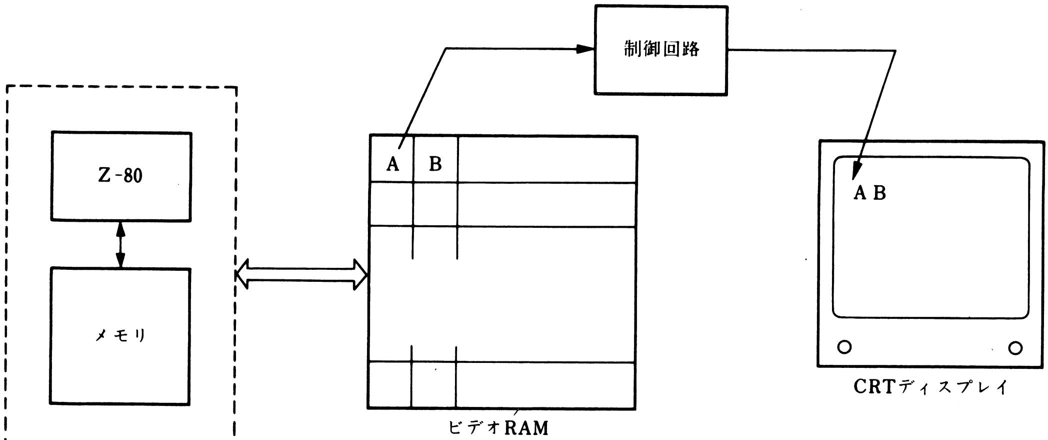


図4.16　キャラクタディスプレイの原理

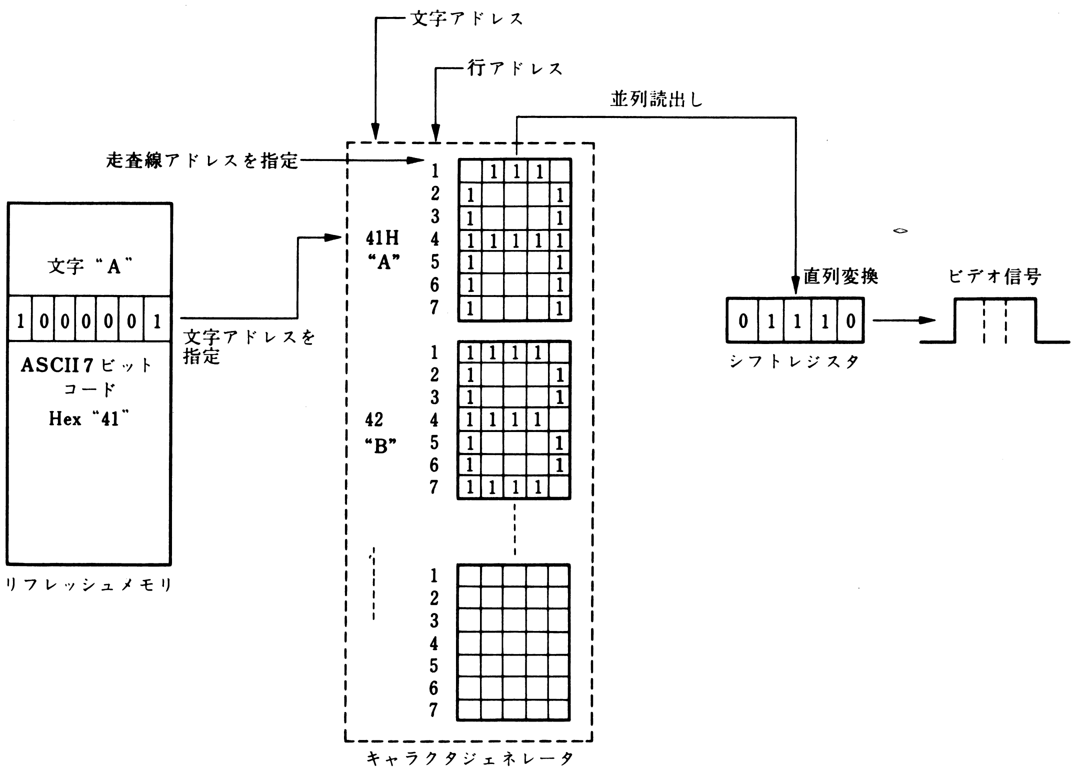


図4.17　キャラクタ発生の原理

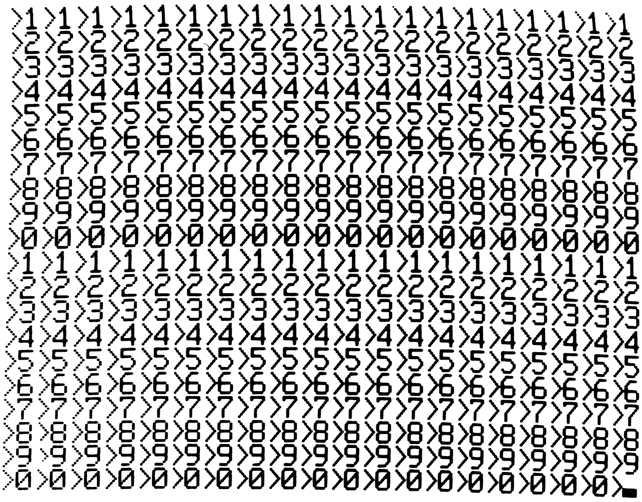


図4.18　キャラクタジェネレータより発生したパターン

ん。そこで，文字コードを文字パターンに変換する必要があります。

この変換に用いるICをキャラクタジェネレータとよんでいます。**図4.17**に，文字コードが文字パターンに変換される原理を示します。

キャラクタジェネレータは，文字コードと走査線の番号とをアドレスとして文字パターンを発生するROMです。通常，文字の1ドットの時間は数十nsと短いため，文字パターン発生は文字ドットの行または列ごとに並列に読み出し，これを並列から直列変換して，CRTに表示する方法を用います。**図4.18**は，キャラクタジェネレータで発生したパターンをCRT画面上に表示した一例です。

電子ビームが面画上を走査するに従って，その走査位置に対応するVRAM上のデータが順次読み出されます。このとき，電子ビームが何本目の走査線であるかを示す信号により，対応する行の文字パターンが出力されます。

4.3.2　CRT制御ICの使い方

CRTディスプレイの制御用には，CRT専用LSIのHD 46505 RP*（以下，46505またはCRTCと略します）を用います。46505はHMCS 6800 LSIファミリの一つで，ラスタスキャン型のCRTコントローラであり，その内部ブロック図を**図4.19**に示します。

CRTCの主要な機能の一つは，マイクロプロセッサZ-80 Aでプログラムされた仕様に従って，ラスタスキャンタイプのCRTディスプレイに必要な水平同期信号や垂直同期信号などの各種タイミング信号を発生することです。

HD 46505では，多くのディスプレイ信号の仕様がプログラマブルであるため，各自のディスプレイの機能に応じたディスプレイ画面が容易に得られることになります。

■HD 46505の特徴

HD 46505は，5V単一電源で動作し，入出力端子はすべてTTLコンパチブルとなっており，40ピンパッケージに入っています。**図4.20**にピン配置を示します（43ページ参照）。

HD 46505は，**表4.2**に示すように，1画面の表示文字数，1文字のドット構成，水平，垂直同期信号などがプログラム可能となっています。またカーソル表示やスタートアドレスがプログラム可能となっているので，ページングやスクローリングが容易にできます。

■HD 46505とZ-80のインターフェース信号

① D_7～D_0は，双方向性データバスでCRTCとZ-80間のデータの送信／受信に使用さ

* HD 46505 RPは3.7MHz高速表示動作型です。SPでも可。

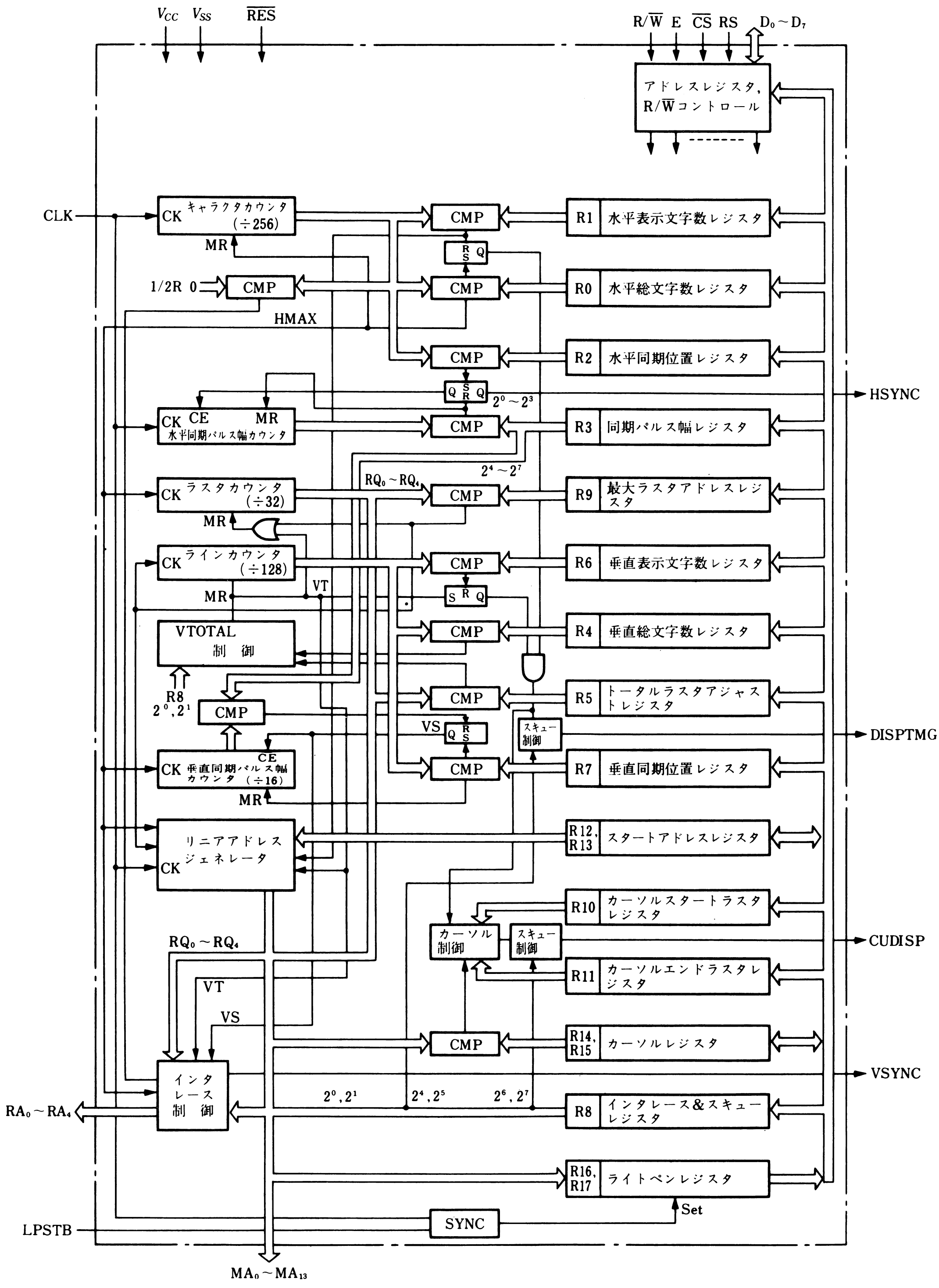


図4.19　HD 46505の内部ブロック図

れます。データバスの出力はトライステートバッファになっており，Z-80がCRTCのデータをリードするとき以外は，ハイインピーダンス状態になっています。

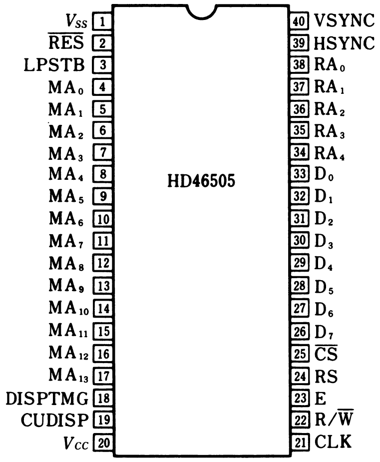


図4.20 HD 46505のピン配置

② リード／ライト$(R/\overline{W})$信号は，Z-80とCRTC間のデータ転送の方向を制御します。すなわち，$R/\overline{W}$端子が"High レベルのときCRTCのデータがZ-80に転送され，"Low"レベルのときZ-80からCRTCにデータが転送されます。

③ $\overline{CS}$はチップセレクト信号であり，$\overline{CS}$が"Low"レベルのときのみCRTCの内部レジスタに対してリード／ライト動作ができます。ここでは，図4.13の配線図内 **IC** に示すように，$\overline{IOW}$信号とアドレスA_6が"Low"レベルで選択されます。手作りコンピュータシステムでは，ライトペンなどは使用しないため，CRTCからZ-80へのデータのリード操作は行なわず，$\overline{IOW}$とのみAND

表4.2 HD 46505の機能

分類	機能	備考
プログラマブルな画面構成	水平走査の周期 垂直走査の周期（行単位） 垂直走査の周期（微調整） 1行の表示文字数 1画面の表示行数 1行のラスタ数 CRT 上の水平方向表示位置 CRT 上の垂直方向表示位置 水平同期信号のパルス幅 垂直同期信号のパルス幅	1文字時間単位でプログラム可能 1行時間単位でプログラム可能 1ラスタ単位でプログラム可能 1文字の縦方向ドット数＋行間隔 同期信号の出力タイミングをプログラムすることにより可能
プログラマブルなカーソル表示	画面上のカーソル表示位置 カーソルの形状 カーソルのブリンクとその周期	14ビットのカーソルレジスタ内蔵 ブリンク周期は16または32フィールド周期のいずれかを選択可能
スキャンモード	ノンインタレースモード インタレースシンクモード インタレースシンクおよびビデオモード	3つのモードのいずれか1つを選択可能
ライトペン	14ビットのライトペンレジスタ内蔵	
リフレッシュメモリアドレッシング機能	14ビットのリフレッシュメモリアドレスを出力	最大16K語のリフレッシュメモリをアクセス可能
プログラマブルなスタートアドレス	14ビットのスタートアドレスレジスタを内蔵（リフレッシュメモリ上の表示開始アドレスをプログラムで指定可能）	ページング，スクローリングなどの応用操作可能

回路を形成しています。

④　24番ピンのRSは，レジスタセレクト信号で，図4.19に示したCRTC内部の1個のアドレスレジスタと，R_0からR_{17}の合計18個のコントロールレジスタを選択するのに使います。すなわち，RSが“Low”レベルのときアドレスレジスタが選択され，“High”レベルのときコントロールレジスタが選択されます。RS信号の使い方については，後で詳しく述べます。

⑤　イネーブル信号（E）は，Z-80がCRTCの内部レジスタをリード／ライトするときのストローブ信号として使用します。このE信号として，ここでは図4.13の**1B**に示すフリップフロップ74 LS 74を用いて，Z-80 CPUのクロック信号から形成しています。

⑥　$\overline{\text{RES}}$はリセット信号で，外部からCRTCをリセットするために使用します。ここでは，CPUをリセットする信号$\overline{\text{RES}}$を直接利用しています。リセット信号が入ると，CRTCの出力信号はすべて“Low”レベルとなりますが，CRTC内部のアドレスレジスタとコントロールレジスタは影響を受けず，前の状態を保持します。

■HD 46505とディスプレイ間の信号

①　キャラクタクロック信号（CLK）は，CRTC内の動作の基準となるクロック入力信号です。ここでは，図4.13の**4E**に示す8 MHzの発振回路の出力を，4ビットカウンタ（74 LS 163）で**図4.21**に示すように，1／8に分周した1 MHzの出力Q_Cを用いています。

②　HSYNCは，水平同期信号で，CRTディスプレイに対して水平同期をかけるための出力信号です。

③　VSYNCは，垂直同期信号で，CRTディスプレイに対して垂直同期をかけるための出力信号です。垂直同期信号のパルス幅は16ラスタ分に固定されています。

④　DISPTMGは，表示タイミング信号です。この信号は，水平および垂直走査の画像表示期間を示します。この信号が“High”レベルのときのみ表示するようになっています。

⑤　MA_0～MA_{13}はVRAMのアドレス出力信号で，14ビットが出力されており，最大16 Kバイトのアドレスが指定できます。

⑥　RA_0～RA_4は，ラスタアドレスでキャラクタジェネレータの行アドレス番号を供給しています。ここでは，RA_0からRA_4までの5ビット出力なので，最大32行までのアドレスが指定可能です。漢字キャラクタジェネレータの制御にも利用できることになります。

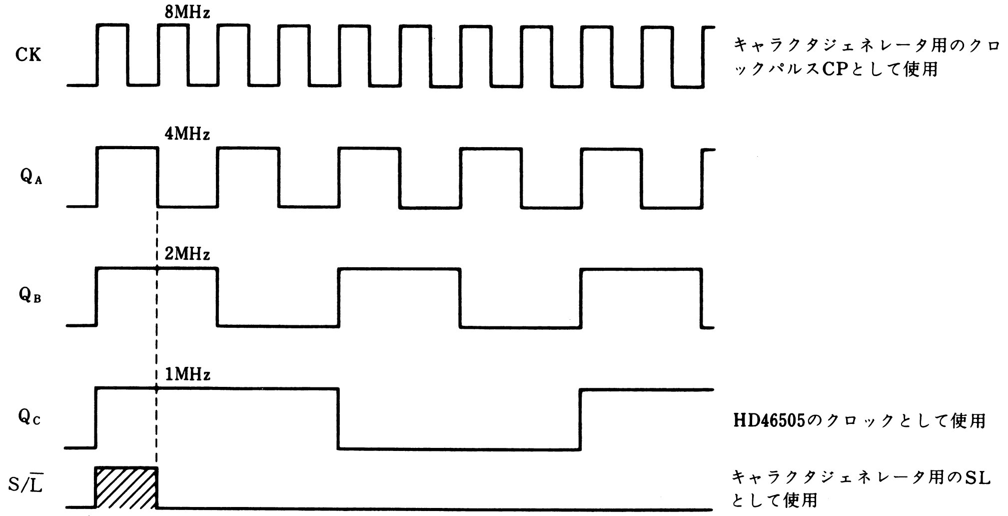


図4.21　4ビットカウンタ74LS163の出力波形

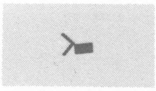


図4.22 カーソルの表示例

⑦ CUDISPはカーソル表示信号で**図4.22**に示すようにCRTディスプレイ画面上にカーソルを表示します。

■内部レジスタの構成と機能

HD 46505には，**表4.3**に示すレジスタがあります。各レジスタの機能は，**図4.23**のCRT画面構成を参照すると理解しやすいでしょう。

① アドレスレジスタ（AR）

このレジスタは，以下に述べるコントロールレジスタ(R 0 ～R 17)のレジスタ番号を指定するレジスタで，R 0 ～R 17にリード／ライトするときには，このアドレスレジスタにコントロールレジスタ番号を書き込む必要があります。RSと$\overline{CS}$が“Low”レベルのときに，このレジスタが選択されます。この条件が表4.3の最左列の$\overline{CS}$とRSで示してあります。

② 水平総文字数レジスタ（R 0）

水平走査の周期を指定します。すなわち水平帰線期間を含めた文字数を指定し，総文字数をNhtとするとき，このレジスタには（Nht－1）を書き込みます。

③ 水平表示文字数レジスタ（R 1）

1行当たりの文字数を指定します。

④ 水平同期位置レジスタ（R 2）

水平同期信号の出力位置を指定します。

⑤ 水平同期パルス幅レジスタ(R 3)

表4.3 HD 46505の内部レジスタ

$\overline{CS}$	RS	アドレスレジスタ					レジスタ	レジスタ名称	プログラム	データビット								備考
		4	3	2	1	0	＃		単位	7	6	5	4	3	2	1	0	
1	×	×	×	×	×	×		無効	—									
0	0	×	×	×	×	×	AR	アドレスレジスタ	—									
0	1	0	0	0	0	0	R 0	水平総文字数*	キャラクタ									(注1)
0	1	0	0	0	0	1	R 1	水平表示文字数	キャラクタ									
0	1	0	0	0	1	0	R 2	水平同期位置*	キャラクタ									
0	1	0	0	0	1	1	R 3	水平同期パルス幅	キャラクタ									
0	1	0	0	1	0	0	R 4	垂直総文字数*	ライン									
0	1	0	0	1	0	1	R 5	トータルラスタアジャスト	ラスタ									
0	1	0	0	1	1	0	R 6	垂直表示文字数	ライン									(注2)
0	1	0	0	1	1	1	R 7	垂直同期位置*	ライン									
0	1	0	1	0	0	0	R 8	インタレースモード	—							V	S	(注3)
0	1	0	1	0	0	1	R 9	最大ラスタアドレス*	ラスタ									(注4)
0	1	0	1	0	1	0	R 10	カーソルスタート	ラスタ		B	P						
0	1	0	1	0	1	1	R 11	カーソルエンド	ラスタ									
0	1	0	1	1	0	0	R 12	スタートアドレス(H)	—									
0	1	0	1	1	0	1	R 13	スタートアドレス(L)	—									
0	1	0	1	1	1	0	R 14	カーソル (H)	—									
0	1	0	1	1	1	1	R 15	カーソル (L)	—									
0	1	1	0	0	0	0	R 16	ライトペン (H)	—									
0	1	1	0	0	0	1	R 17	ライトペン (L)	—									

(注) 1) インタレース（R 8＝1または3）のとき，設定データは奇数であること
2) R8＝3のとき，N/2(N：実際の表示行数)
3) VはSが1のときビデオモードを指定する。Sはインタレースモードを指定する。
4) R 8＝3のとき，設定データは奇数であること。
* (設定値)＝(指定値)－1のレジスタを示す。

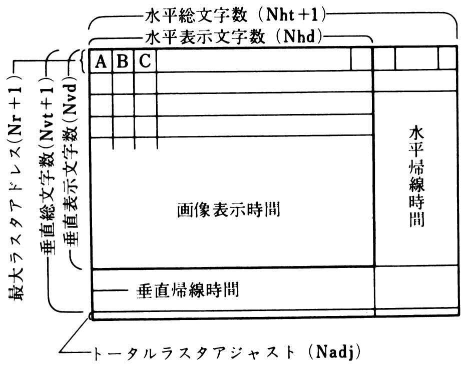


図4.23　CRT画面構成

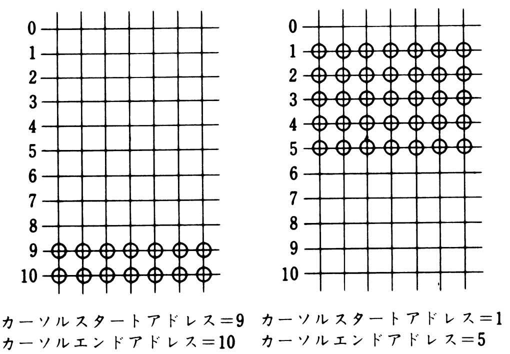


図4.24　カーソル位置の設定例

表4.4　カーソル表示モード

2^6	2^5	カーソル表示モード
0	0	カーソルは点滅しない
0	1	カーソルは表示されない
1	0	カーソルは16フィールド時間で点滅する
1	1	カーソルは32フィールド時間で点滅する

水平同期パルス幅を指定します(1～15の間)。

⑥　垂直総文字レジスタ (R 4)

垂直走査の同期を指定します。

⑦　トータルラスタアジャストレジスタ (R 5)

総ラスタ数を調整するため，1フィールドの最後に付加するラスタの数です。

⑧　垂直表示文字数レジスタ (R 6)，垂直同期位置レジスタ (R 7)

これらのレジスタで，垂直文字の行数と垂直同期信号の位置を指定します。

⑨　インタレースモードレジスタ(R 8)

"00"でノンインタレースモードを，また"01"でインタレースモードを指定します。

⑩　最大ラスタアドレスレジスタ (R 9)

行間のスペースを含めた1行のラスタ数を指定します。

⑪　カーソルスタートレジスタ(R 10)とカーソルエンドレジスタ(R 11)

このレジスタの下位5ビットにカーソルの開始ラスタ行番号を設置し，カーソルエンドレジスタで終了行番号を指定します。**図4.24**にその指定例を示します。また，カーソルスタートアドレスの上位2ビットで，**表4.4**に示すカーソル表示モードが指定できます。

⑫　スタートアドレスレジスタ(R12, R13)

VRAMの読出し先頭アドレスを指定します。

⑬　カーソルレジスタ (R 14，R 15)

カーソルが表示すべき文字に相当するVRAMのアドレスを指定します。

4.3.3　CRT制御基板の動作

図4.13に，HD 46505を用いた配線図を示しましたが，このCRT制御基板の動作について述べます。

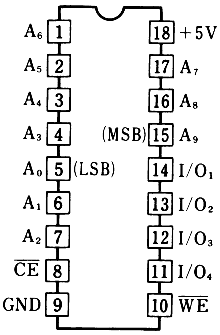
■VRAM

VRAMは図4.13上の**4C**の2つの4KビットスタティックRAM 2114を用います。この2114は**図4.25**に示すピン配置をしており，2つで1Kバイト（0～3 FFH番地）のメモリとなっています。

VRAMのアドレス設定は，図4.13上の**1A**と**2A**に配置した2つのTTL 74 LS 42で行なっています。B 1の74 LS 42の出力は $\overline{7}$ を用いていますので，$\overline{MEMW}$ が "Low" で，かつ A_{15}，A_{14}，A_{13} がすべて"High"の条件を定め，B 11の74 LS 42は出力4を用います。したがって A_{12} のみが"High"の条件を定めています。

これより，F 000 H番地からVRAMスタートアドレスとなり，F 000 H～F 3 FFHがVRAM領域となります。

一方，ラスタスキャンディスプレイは，た



入 力		I/O	動 作
$\overline{CE}$	$\overline{WE}$		
H	X	High-Z	非選択
L	L	L	Write"L"
L	L	H	Write"H"
L	H	DO	Read

図4.25 2Kスティック RAM2114のピン配置とその動作表

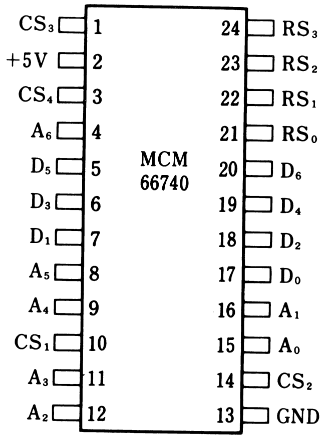


図4.26 モトローラ製キャラクタジェネレータ MCM66740のピン配置

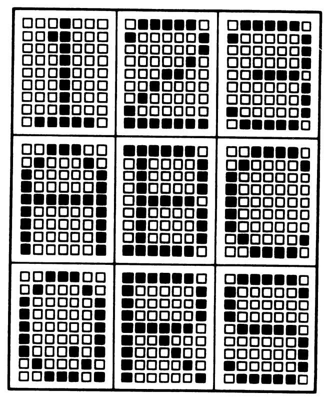


図4.27 MCM66740の文字発生パターン（部分）

えずリフレッシュを行ないます。すなわち，たえずVRAM 2114を読み出す作業をしていますので，Z-80のアドレスやデータバスと競合する恐れがあります。そこで，Z-80がVRAMにデータを書き込むときだけアドレスバスとデータを管理し，それ以外の場合は切り離し，CRTCのアドレスMA_0～MA_9に従ってVRAMが動作するようにしなければなりません。

このためのバスラインの選択を行なうのが図4.13上**2C**，**2D**，**2E**にある3つのデータセレクタ用TTL-IC LS157であり，**7B**にあるLS244です。

■**キャラクタジェネレータ**

キャラクタジェネレータは，アスキー*コードで小文字が発生できるモトローラ社製MCM66740を用います。キャラクタジェネレータのピン配置を**図4.26**に示します。このキャラクタジェネレータは**図4.27**に示すように7×9ドットで1文字を表示します。

図4.13上**5C**に示すように，行指定RS_0～RS_3をCRTCのRA_0～RA_3に配線し，文字コードを指定するA_0～A_6をVRAMのデータバスとすることにより，出力データD_0～D_7に文字パターンが得られます。

■**2716によるキャラクタジェネレータの作製**

モトローラ社製の小文字が発生できるMCM66740は，高価でかつ入手しにくいものです。そこで2KバイトのEPROMである2716を用いて，キャラクタジェネレータを作る方法を紹介しましょう。

2716を用いたキャラクタジェネレータのダンプリストの例を**表4.5**に，また**図4.28**に表示文字を示します。

2716のEPROMライタをもっている方や第6章で作製する手動のEPROMライタで書き込む元気のある読者は，キャラクタジェネレータとして2716を用いることをおすすめします。もし，2716を利用する場合はキャラクタジェネレータの部分の配線を，**図4.29**に示すように変更して下さい。

* ASCII (American Standard Code for Information Interchange)

表4.5 2716を用いたキャラクタジェネレータのダンプリスト

0000	00	00	00	00	00	00	00	00
0008	00	00	00	00	00	00	00	00
0010	00	00	00	00	00	00	00	00
0018	00	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00
0028	00	00	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	00
0038	00	00	00	00	00	00	00	00
0040	00	00	00	00	00	00	00	00
0048	00	00	00	00	00	00	00	00
0050	00	00	00	00	00	00	00	00
0058	00	00	00	00	00	00	00	00
0060	00	00	00	00	00	00	00	00
0068	00	00	00	00	00	00	00	00
0070	00	00	00	00	00	00	00	00
0078	00	00	00	00	00	00	00	00
0080	00	00	00	00	00	00	00	00
0088	00	00	00	00	00	00	00	00
0090	00	00	00	00	00	00	00	00
0098	00	00	00	00	00	00	00	00
00A0	00	00	00	00	00	00	00	00
00A8	00	00	00	00	00	00	00	00
00B0	00	00	00	00	00	00	00	00
00B8	00	00	00	00	00	00	00	00
00C0	00	00	00	00	00	00	00	00
00C8	00	00	00	00	00	00	00	00
00D0	00	00	00	00	00	00	00	00
00D8	00	00	00	00	00	00	00	00
00E0	00	00	00	00	00	00	00	00
00E8	00	00	00	00	00	00	00	00
00F0	00	00	00	00	00	00	00	00
00F8	00	00	00	00	00	00	00	00
0100	00	00	00	00	00	00	00	00
0108	00	04	04	04	04	04	00	04
0110	00	0A	0A	0A	00	00	00	00
0118	00	0A	0A	1F	0A	1F	0A	0A
0120	00	04	0F	14	0E	05	1E	04
0128	00	18	19	02	04	08	13	03
0130	00	08	14	14	08	15	12	0D
0138	00	04	04	04	00	00	00	00
0140	00	04	08	10	10	10	08	04
0148	00	04	02	01	01	01	02	04
0150	00	04	15	0E	04	0E	15	04
0158	00	00	04	04	1F	04	04	00
0160	00	00	00	00	00	04	04	08
0168	00	00	00	00	1F	00	00	00
0170	00	00	00	00	00	00	00	04
0178	00	00	01	02	04	08	10	00
0180	00	0E	11	13	15	19	11	0E
0188	00	04	0C	04	04	04	04	0E
0190	00	0E	11	01	06	08	10	1F
0198	00	1F	01	02	06	01	11	0E
01A0	00	02	06	0A	12	1F	02	02
01A8	00	1F	10	1E	01	01	11	0E
01B0	00	07	08	10	1E	11	11	0E
01B8	00	1F	01	02	04	08	08	08
01C0	00	0E	11	11	0E	11	11	0E
01C8	00	0E	11	11	0F	01	02	1C
01D0	00	00	00	04	00	04	00	00
01D8	00	00	00	04	00	04	04	08
01E0	00	02	04	08	10	08	04	02
01E8	00	00	00	1F	00	1F	00	00
01F0	00	08	04	02	01	02	04	08
01F8	00	0E	11	02	04	04	00	04
0200	00	0E	11	15	17	16	10	0F
0208	00	04	0A	11	11	1F	11	11
0210	00	1E	11	11	1E	11	11	1E
0218	00	0E	11	10	10	10	11	0E
0220	00	1E	11	11	11	11	11	1E
0228	00	1F	10	10	1E	10	10	1F
0230	00	1F	10	10	1E	10	10	10
0238	00	0F	10	10	10	13	11	0F
0240	00	11	11	11	1F	11	11	11
0248	00	0E	04	04	04	04	04	0E
0250	00	01	01	01	01	01	11	0E
0258	00	11	12	14	18	14	12	11
0260	00	10	10	10	10	10	10	1F
0268	00	11	1B	15	15	11	11	11
0270	00	11	11	19	15	13	11	11
0278	00	0E	11	11	11	11	11	0E
0280	00	1E	11	11	1E	10	10	10
0288	00	0E	11	11	11	15	12	0D
0290	00	1E	11	11	1E	14	12	11
0298	00	0E	11	10	0E	01	11	0E
02A0	00	1F	04	04	04	04	04	04
02A8	00	11	11	11	11	11	11	0E
02B0	00	11	11	11	11	11	0A	04
02B8	00	11	11	11	15	15	1B	11
02C0	00	11	11	0A	04	0A	11	11
02C8	00	11	11	0A	04	04	04	04
02D0	00	1F	01	02	04	08	10	1F
02D8	00	1F	18	18	18	18	18	1F
02E0	00	11	0A	04	0E	04	0E	04
02E8	00	1F	03	03	03	03	03	1F
02F0	00	00	00	04	0A	11	00	00
02F8	00	00	00	00	00	00	00	1F
0300	00	08	08	04	00	00	00	00
0308	00	00	00	0C	02	0E	12	0D
0310	00	10	10	16	19	11	19	16
0318	00	00	00	0C	12	10	12	0C
0320	00	01	01	0D	13	11	13	0D
0328	00	00	00	0E	11	1E	10	0E
0330	00	02	04	04	1F	04	04	04
0338	00	0E	11	11	0F	01	11	0E
0340	00	10	10	10	1C	12	12	12
0348	00	04	00	0C	04	04	04	0E
0350	00	01	00	01	01	01	11	0E
0358	00	10	11	12	14	1A	11	11
0360	00	0C	04	04	04	04	04	0E
0368	00	00	00	1A	15	15	15	15
0370	00	00	00	16	19	11	11	11
0378	00	00	00	0E	11	11	11	0E
0380	00	00	00	1E	11	1E	10	10
0388	00	00	0E	11	0F	01	01	01
0390	00	00	00	16	19	10	10	10
0398	00	0E	11	08	04	02	11	0E
03A0	00	08	08	1E	08	08	09	06
03A8	00	00	00	11	11	11	13	0D
03B0	00	00	00	11	11	11	0A	04
03B8	00	00	00	15	15	15	15	0A
03C0	00	00	00	11	0A	04	0A	11
03C8	00	11	11	11	0F	01	11	0E
03D0	00	00	00	1F	02	04	08	1F
03D8	00	02	04	04	08	04	04	02
03E0	00	04	04	04	04	04	04	04
03E8	00	08	04	04	02	04	04	08
03F0	00	08	15	02	00	00	00	00
03F8	1F	1F	1F	1F	1F	1F	1F	1F

0400	00	00	00	00	00	00	00	00
0408	00	00	00	00	00	00	00	00
0410	00	00	00	00	00	00	00	00
0418	00	00	00	00	00	00	00	00
0420	00	00	00	00	00	00	00	00
0428	00	00	00	00	00	00	00	00
0430	18	3C	3C	3C	3C	7E	FF	FF
0438	DB	FF	DB	18	18	DB	FF	DB
0440	18	5A	5A	5E	58	78	18	18
0448	08	10	08	10	08	10	08	10
0450	3C	3C	18	FF	18	18	24	42
0458	3C	77	FF	FF	FF	FF	7E	3C
0460	3C	42	81	81	81	81	42	3C
0468	81	42	24	00	00	24	42	81
0470	18	18	18	FF	18	18	3C	18
0478	00	00	18	18	FF	FF	FF	FF
0480	18	3C	18	7E	FF	7E	18	3C
0488	18	3C	7E	FF	FF	7E	3C	18
0490	66	FF	FF	FF	7E	3C	18	18
0498	18	18	3C	7E	FF	FF	5A	18
04A0	00	00	00	00	00	00	00	00
04A8	00	00	00	00	00	00	00	00
04B0	00	00	00	00	00	00	00	00
04B8	00	00	00	00	00	00	00	00
04C0	00	00	00	00	00	00	00	00
04C8	00	00	00	00	00	00	00	00
04D0	00	00	00	00	00	00	00	00
04D8	00	00	00	00	00	00	00	00
04E0	00	00	00	00	00	00	00	00
04E8	00	00	00	00	00	00	00	00
04F0	00	00	00	00	00	00	00	00
04F8	00	00	00	00	00	00	00	00
0500	00	00	00	00	00	00	00	00
0508	00	00	00	00	00	08	14	08
0510	00	0F	08	08	08	00	00	00
0518	00	00	00	00	02	02	02	1E
0520	00	00	00	00	00	08	08	10
0528	00	00	00	00	04	00	00	00
0530	00	00	1F	01	1F	01	02	04
0538	00	00	00	0F	01	05	04	08
0540	00	00	00	02	04	0C	14	04
0548	00	00	00	08	1E	12	02	04
0550	00	00	00	00	0E	04	04	1F
0558	00	00	00	04	1E	04	0C	14
0560	00	00	00	08	1F	09	0A	08
0568	00	00	00	00	0E	02	02	1F
0570	00	00	00	0F	01	0F	01	0F
0578	00	00	00	00	15	15	01	06
0580	00	00	00	00	1F	00	00	00
0588	00	00	1F	01	05	04	04	08
0590	00	02	04	0C	14	04	04	04
0598	00	04	1F	11	01	02	04	08
05A0	00	00	1F	04	04	04	04	1F
05A8	00	02	1F	02	06	0A	12	02
05B0	00	04	04	1F	05	05	09	12
05B8	00	04	1F	04	1F	04	04	04
05C0	00	00	0F	09	11	02	04	08
05C8	00	08	08	0F	12	02	02	04
05D0	00	00	1F	01	01	01	01	1F
05D8	00	00	0A	1F	0A	0A	02	04
05E0	00	00	18	00	19	01	02	1C
05E8	00	00	1F	01	02	04	0A	11
05F0	00	00	08	1F	09	08	08	0F
05F8	00	00	11	09	01	01	02	04
0600	00	00	0F	09	15	02	04	08
0608	00	02	1C	04	1F	04	04	08
0610	00	00	15	15	01	01	02	04
0618	00	00	0E	00	1F	04	04	08
0620	00	08	08	08	0C	0A	08	08
0628	00	04	04	1F	04	04	04	08
0630	00	00	0E	00	00	00	00	1F
0638	00	00	1F	01	0A	04	0A	10
0640	00	04	1F	02	06	0D	14	04
0648	00	00	01	01	02	04	08	10
0650	00	00	00	0A	09	09	09	11
0658	00	00	10	10	1F	10	10	1F
0660	00	00	1F	01	01	02	04	08
0668	00	00	00	08	14	02	01	00
0670	00	04	04	1F	04	15	15	04
0678	00	00	1F	01	01	05	04	02
0680	00	08	07	00	07	00	08	07
0688	00	00	02	04	08	11	1F	01
0690	00	00	01	01	0A	04	0A	10
0698	00	00	1E	08	1F	08	08	0F
06A0	00	08	08	1F	09	0A	08	08
06A8	00	00	0E	02	02	02	02	1F
06B0	00	00	1F	01	0F	01	01	1F
06B8	00	00	0E	00	1F	01	01	02
06C0	00	00	09	09	09	01	01	02
06C8	00	00	14	14	14	15	15	16
06D0	00	00	10	10	11	12	14	18
06D8	00	00	1F	11	11	11	11	1F
06E0	00	00	1F	11	11	01	02	04
06E8	00	00	18	01	01	02	04	18
06F0	00	14	14	00	00	00	00	00
06F8	00	08	14	08	00	00	00	00
0700	00	00	00	18	3C	7E	FF	FF
0708	FF	FF	7E	3C	18	00	00	00
0710	00	00	00	18	24	42	81	81
0718	81	81	42	24	18	00	00	00
0720	AA	55	AA	55	AA	55	AA	55
0728	10	10	10	10	1F	00	00	00
0730	10	10	10	10	F0	00	00	00
0738	00	00	00	00	F0	10	10	10
0740	00	00	00	00	1F	10	10	10
0748	10	10	10	10	10	10	10	10
0750	00	00	00	00	FF	00	00	00
0758	10	10	10	10	FF	10	10	10
0760	01	02	04	08	10	20	40	80
0768	80	40	20	10	08	04	02	01
0770	81	C3	E7	FF	FF	E7	C3	81
0778	FF	7E	3C	18	18	3C	7E	FF
0780	00	00	00	00	00	00	00	00
0788	FF	FF	FF	FF	F0	F0	F0	F0
0790	F0	F0	F0	F0	0F	0F	0F	0F
0798	00	00	00	00	F0	F0	F0	F0
07A0	0F	0F	0F	0F	0F	0F	0F	0F
07A8	F0	F0	F0	F0	FF	FF	FF	FF
07B0	F0	F0	F0	F0	00	00	00	00
07B8	0F	0F	0F	0F	FF	FF	FF	FF
07C0	0F	0F	0F	0F	00	00	00	00
07C8	F0	F0	F0	F0	F0	F0	F0	F0
07D0	FF	FF	FF	FF	0F	0F	0F	0F
07D8	0F	0F	0F	0F	F0	F0	F0	F0
07E0	00	00	00	00	0F	0F	0F	0F
07E8	FF	FF	FF	FF	FF	FF	FF	FF
07F0	FF	FF	FF	FF	00	00	00	00
07F8	00	00	00	00	FF	FF	FF	FF

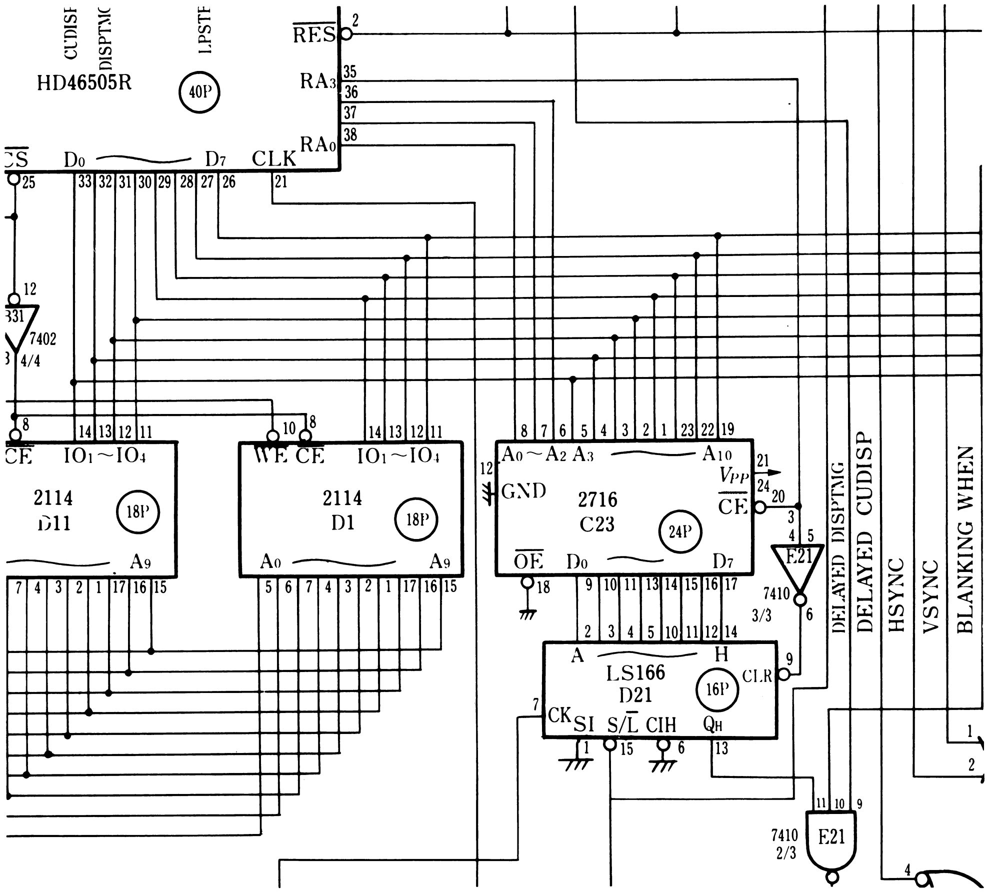


図4.29　EPROM 2716を用いたときのキャラクタジェネレータ回路の配線図

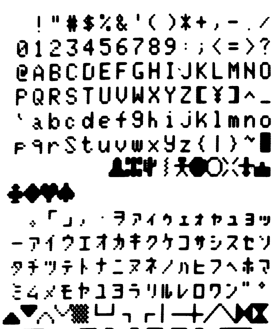
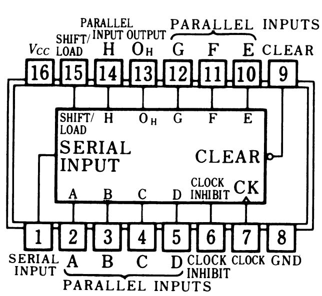


図4.28　キャラクタジェネレータの文字パターン



入	力			動　作
Clear	Shift/Load	CK	CK Inhibit	
H	H	↑	L	右シフト
H	L		L	ロード
H	H	×	H	ホールド
⊔	×	×	×	クリア

図4.30　TTL 74166のピン配置と動作

■並列─直列変換

図4.17に示したようにキャラクタジェネレータにより同時に8ビットの並列パターンとして得られたデータを時系列信号に直します。すなわち，8ビット並列を直列に変換するのが図4.13上**6D**に配置するシフトレジスタTTL 74 LS 166です。

74 LS 166ピン配置を**図4.30**に示します。このシフトレジスタに対する追出し用のクロック

パルスCPは，図 4.21 に示すように，HD 46505 のクロック周波数 1 MHzに対して 8 倍の周波数 8 MHzの周波数を用います。また，S/$\overline{L}$はシフトレジスタへのデータ取込み用の入力信号であり，図 4.13 上**5E**の 3 入力NANDで構成し，その波形は図 4.21 の最下行に示す 1 MHzの短いパルスを用います。

■ビデオ制御

HD46505からはビデオ制御信号として，水平同期信号 HSYNC，垂直同期信号 VSYNC のほかに，ビデオ信号のブランク期間を制御する DISPTMG信号およびカーソルを表示するための CUDISP 信号があります。DISPTMGとCUDISP信号に対し，並列―直列変換された文字パターンとのタイミングをとるために，図4.13上**5A**，**6A**にあるフリップフロップTTL74LS74が挿入されています。

また，同図**7D**，**8D**にあるビデオ信号発生回路は，個々に発生しているHSYNC, VSYNCと映像信号との混合信号を得るための回路であり，**図 4.31** に示すような映像信号がトランジスタの最終出力として求まります。この信号を，ビデオ端子入力のあるテレビまたはディスプレイに入れることにより，画像が得られます。

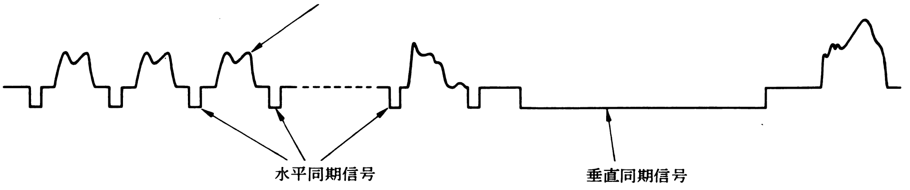


図4.31　複合映像信号の例

4.4 64Kメモリ基板の作り方

第 4 章　基本回路の作り方

現在安価に出まわりはじめ，かつ性能が安定しており，比較的手軽に利用できるようになった 64 Kダイナミック RAMを 8 個使用して，64 Kバイトの主記憶メモリを作製します。64 Kバイトは，Z-80 の全アドレス空間をカバーしており，1 世代前の小型計算機のメモリ容量に匹敵しています。

はじめに 64 Kダイナミック RAM 4164 の性能と特徴について紹介し，次に作製する 64 Kメモリ基板の動作について示します。

4.4.1　64 K DRAMの使い方

64 K　DRAMは，標準 16 ピンパッケージで 5 V単一電源で動作する非常に使いやすいICです。64 K DRAMはメーカによって異なる型名を使用していますので，**表 4.6** にメーカ別の DRAM一覧表をあげておきます。

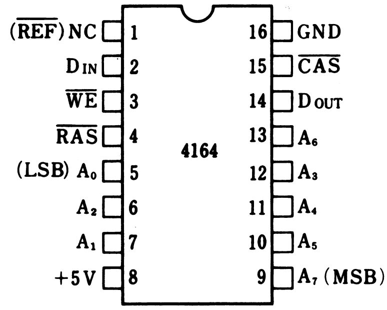
本書では，64 K DRAMを一般に広く知られている 4164 とよぶことにします。また，64 K DRAMの使用法や設計に関する多くの解説がありますので，[4.3)～4.5)]ここでは簡単に 4164 の特性について述べます。

■DRAM 4164 の電気的特性[4～3)]

DRAM 4164 は，**図 4.32** に示すピン接続の標準 16 ピンパッケージで，65,536× 1 ビット構成となっています。アクセス時間は 150／200 ns で，最大 28 mWの低消費型です。また 128 リフレッシュサイクル（A_0～A_6）のものがほとんど

表4.6　64K DRAM一覧

型　名	メーカ	アクセスタイム t_{RAC}(ns)					t_{RCD}/t_{CAC}	リフレッシュ	1ピンリフ
		100	120	150	200	250	(t_{RAC}=150ns)	サイクル/ms	レッシュ
MB 8264	富士通			○	○		50／100	128／2	×
MB 8265	富士通			○	○		50／100	128／2	○
μPD4164	日　電			○	○	○	50／100	128／2	×
HM 4864	日　立			○	○		50／100	128／2	×
TMM 4164	東　芝			○	○		50／100	128／2	×
MSK 4164	三　菱			○	○		75／75	128／2	○
SM 3764	沖		○	○	○		50／100	128／2	×
TMS 4164	T　I			○	○	○	／	256／4	×
MCM 6664	モトローラ			○	○	○	75／75	128／2	○
MCM 6665	モトローラ			○	○	○	75／75	128／2	×
NMC 4164	N　S		○	○	○		／	256／4	×
2164	インテル			○	○	○	65／85	128／2	×
64 K	フェアチャイルド		○	○	○		／	256／4	×
MK4164	モステック	○	○	○	○		／	／	○



記　号	端　子　名　称
$A_0 \sim A_7$	アドレス入力
$\overline{RAS}$	ロウ アドレス ストローブ
$\overline{CAS}$	カラム アドレス ストローブ
$\overline{WE}$	ライト イネーブル
D_{IN}	データ入力
D_{OUT}	データ出力
NC	ノー コネクション

図4.32　DRAM 4164のピン接続

で，非常に使いやすくなっています。

4164の1番ピンについては，図4.32ではNC（No Conection）未使用となっていますが，三菱，モトローラなどは1番ピンによるオート／セルフリフレッシュ機能が採用されています。各入出力端子は，TTLコンパチブルであり，ファンアウトもLS-TTLで10.5個と十分なドライブ能力があります。

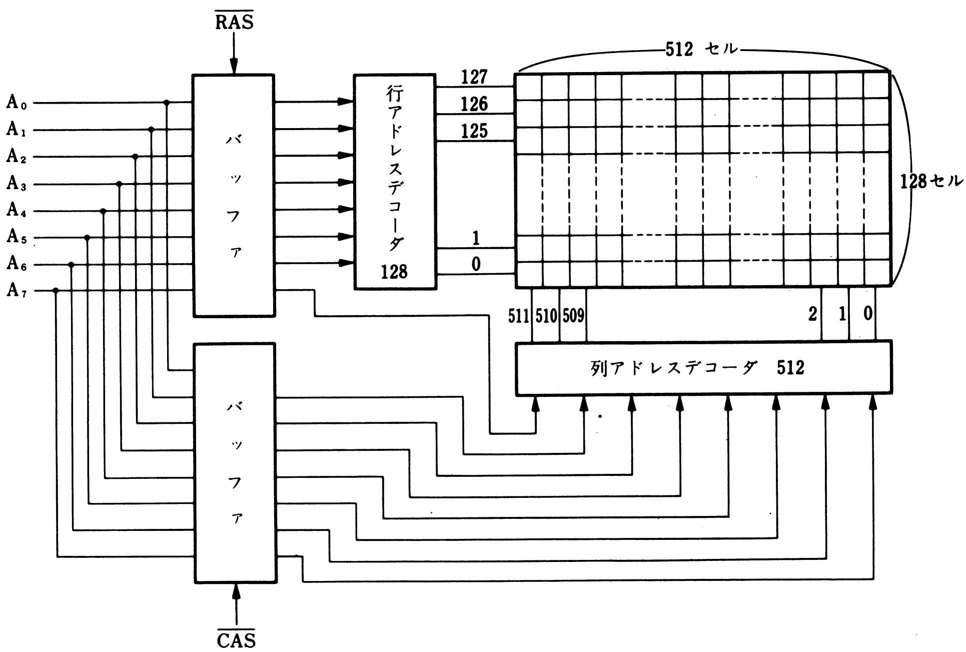


図4.33　64K DRAM (4164)のメモリアレイ

■ **128 リフレッシュサイクル**

DRAMはセル構造をしており，64 Kビットが**図 4.33** に示すように，128×512 の矩形格子状構造となっています。リフレッシュは図中の行単位に行なわれます。

すなわち，128 の中の一つの行を指定することにより，512 個の素子が同時にリフレッシュされます。このような構成を **128 リフレッシュアドレス構成**とよびます。

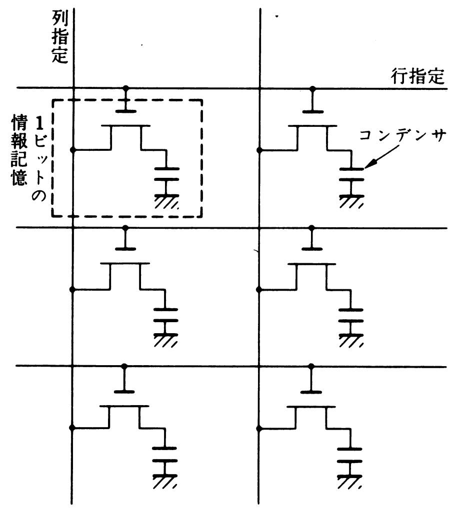
DRAMの内容が消滅してしまわない 2 ms以内に 128 回のリフレッシュ回数が必要となります。そのため，リフレッシュ間隔は 15.6 μs(＝

《ダイナミックRAMとは[4-4]》

スタティックRAMは，1ビットの情報を一つのフリップフロップのONとOFFに対応させて記憶します。そのため，記憶された状態は，電源を切らない限り保存されます。それに対し，ダイナミックRAMは，一つのコンデンサの充電状態と放電状態に対応させて，1ビットの情報を記憶します。

また，ダイナミックRAMの容量は約 0.07 pFと小さく，データを書き込んだ後，時間の経過とともにその書き込まれたデータが消滅します。これはコンデンサのリーク電流によって放電するためです。

ダイナミックRAMは，**付図 4.5** に示すセル構造をしており，行アドレスと列アドレスを指定することにより，1ビットに対応するコンデンサが記憶動作を行ないます。



付図4.5 64K DRAMのセル構造

《リフレッシュとは》

DRAMは，一度書き込まれたデータが完全に放電してしまう前に，再度同じデータをセルに注入することを繰り返すことにより，データが保持されます。このように，充電状態にあるセルに対しては，ふたたび電荷を注入する動作をリフレッシュといいます。

データを正しく保持するために必要なリフレッシュ間隔は，ICにより異なりますが，普通は 2 ms のものが多いようです。

《オート／セルフリフレッシュ》

128 セルフリフレッシュのリフレッシュアドレスは，Z-80 の機能としてあるリフレッシュアドレスレジスタより発生される信号を利用します。それに対し，オート／セルフリフレッシュ機能というのは，4164 DRAM内にリフレッシュアドレスカウンタ回路，リフレッシュアドレスレジスタ回路などが含まれており，外部回路なしでセルフリフレッシュ動作が行なえる素子です，

セルフリフレッシュでは，1 番ピン $\overline{\mathrm{REF}}$ の“Low”レベルが 2 ms以上続くと，内部から同期信号が出て，自動的にリフレッシュ動作を行ないます。

2 ms/128）となります。

Z-80は，命令を1ステップ実行するたびに1回のリフレッシュを行ないます(M1サイクルの後半)。すなわち，1回のリフレッシュごとに，8ビットのリフレッシュカウンタを進めて，リフレッシュアドレスが指定され，アドレスバスA_0～A_7に出力されます。

4 MHzのクロックでZ-80を動作させると，サイクル時間が0.25 μsとなりますので，1命令は5.75 μsで，十分終了することになり，正しくリフレッシュ動作が行なわれることになります。CPUがウエイト状態*にあると，リフレッシュ信号が出なくなります。したがって，ウエイトの時間は2 ms以内に128サイクルの条件を満たすようにしなくてはなりません。

■アドレス指定

64 K DRAMは65536×1ビット構成なので，16本のアドレス指定が必要です。図4.32に示したように4164はA_0～A_7の8本のアドレスラインなので，2回に分けて指定します。

図4.34に示すように，下位8ビットの行アドレス情報が行アドレス信号である$\overline{RAS}$ (Row Address Signal) クロックの立ち下がりにより，行バッファに取り込まれます。同様にして，上位8ビットの列アドレス情報**が列アドレス信号である$\overline{CAS}$ (Column Address Signal) クロックにより，列バッファに取り込まれます。同図に示すように，最初に$\overline{RAS}$信号により下位8ビットが，また約30 ns遅れて$\overline{CAS}$信号により上位8ビットがセットされます。

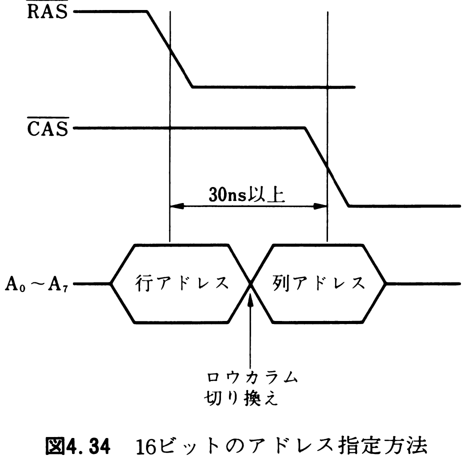


図4.34　16ビットのアドレス指定方法

■READ/$\overline{WRITE}$動作

$\overline{RAS}$信号と同時に下位8ビットのアドレスが指定された後，**図4.35(a)**に示すように，$\overline{CAS}$信号により上位8ビットのアドレスが指定されたとき，$\overline{WE}$ (Write Enable) が"High"レベルであれば，リード動作，すなわちDRAMの内容がDO (Data Out) 端子に出力されます。

また，図4.35 (b) に示すように，$\overline{WE}$端子が"Low"レベルのときはWRITE動作，すなわちDI (Dada In) 端子のデータがDRAMに記憶されます。

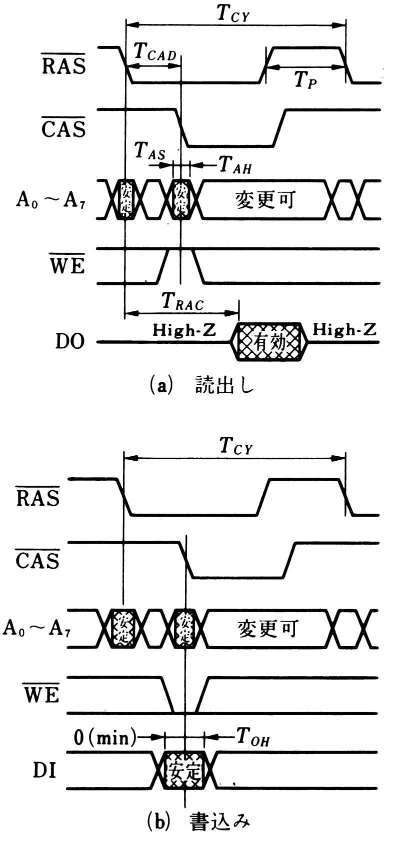


図4.35　リード／ライト動作

* $\overline{WAIT}$動作および$\overline{BUSREQ}$信号によるDMA動作時には，リフレッシュ動作も停止するので，$\overline{WAIT}$と$\overline{BUSREQ}$信号を2 ms以上"Low"レベルにしないようにしなければなりません。

** 図4.33に示すように64 K DRAMは128×512の行列構成なので，正確には行の残り1ビットと7ビットの列アドレスが指定されます。

4.4.2　64 Kメモリ基板の動作

64 K DRAM 4164を8個使用して，64 Kバイトのメモリ基板を製作します。**図 4.36**（56～57ページ）に，製作する64 Kバイトメモリ基板の配線図を示します。

■アドレス切換え

$\overline{RAS}$と$\overline{CAS}$の2回に分けて，A_0～A_{15}の合計16ビットのアドレスを指定します。この下位アドレスと上位アドレスの切換え動作を行なうのが，図4.36の図面中**2B**，**2C**にある4ビットのデータセレクタTTL 74 LS 157です。

74 LS 157は，**図 4.37**に示す動作表からS（Select）が“Low”でA，“High”でBが選択されます。そのため，$\overline{RAS}$信号が出力されるときは$\overline{CAS}$が“High”，すなわちSelect入力のCASが“Low”となっているため，A入力つまり下位アドレスが出力されます。

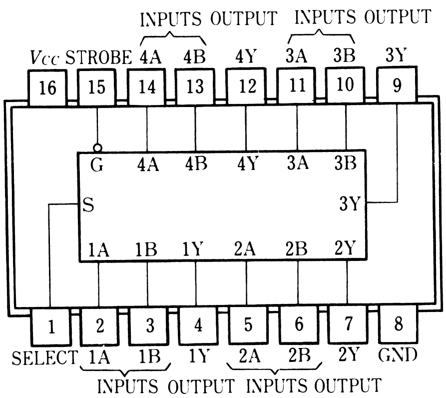
また$\overline{RAS}$信号は，図4.1のCPU基板に示すように，MREQ（Memory Request）信号を用いています。

ここで，図4.36の図面中**2D**にある$\overline{RAS}$信号に対して，2つのNOT回路を通過させているのは，**図 4.38**に示す理由によります。すなわち，$\overline{RAS}$信号が“Low”に下がるとき，すでに下位アドレスが安定していなければならないので，**その安定化の時間を，2つのNOT回路を通過させることにより作っているわけです。**

CAS信号に対するNOT回路3つを通過させているのも，同様な理由です。この場合，2つのNOT回路でもいいわけですが，CASを$\overline{CAS}$にするため，さらに一つ多くNOT回路を通過させています。

■$\overline{CAS}$信号の発生

$\overline{CAS}$信号は，図4.1の図面上**3F**にあるフリップフロップ74 LS 74の出力$\overline{Q}$として得られます。この74 LS 74はパルスの立ち上がりで動作しますので，$\overline{RAS}$信号，すなわち$\overline{MREQ}$信号が発生してから，最大1クロック時間（4 MHz, 250 ns）遅れてから$\overline{CAS}$信号が出力され，データセレクタ74 LS 157により，下位ビットから上位ビットの切換えが行なわれます。この



INPUTS		OUTPUT
Select	Strobe $\overline{G}$	Y
×	High	Low
Low	Low	A
High	Low	B

図 4.37　TTL 74 LS 157のピン配置と動作

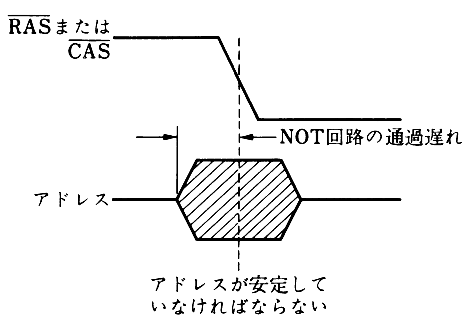


図4.38　NOT回路を挿入する理由

様子を**図 4.39**に示します。

すなわち，$\overline{Q}$信号で行（R）／列（C）の切換えが行なわれ，アドレスが安定した後，$\overline{CAS}$信号が発生することになります。

また，図4.1の図面上**3F**のフリップフロップ74 LS 74の$\overline{PR}$（PReset）入力として$\overline{RFSH}$

が挿入されているのは，リフレッシュ動作中，$\overline{\text{RFSH}}$が“Low”のときはリセットをかけて，$\overline{\text{CAS}}$信号を発生させないためです。つまり，リード／ライト動作は，リフレッシュ動作（$\overline{\text{RFSH}}$が“Low”）でないときに行なわれることになります。

■RASオンリーリフレッシュ

セルフリフレッシュ機能をもたないDRAM 4164（1番ピンがNC）を使用した場合，リフレッシュ動作が必要となります。リフレッ

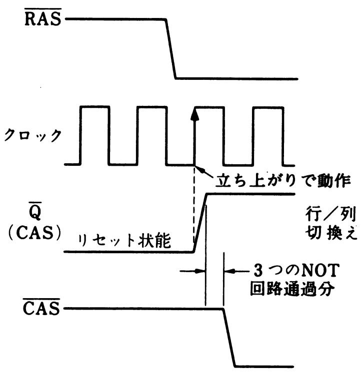


図4.39　$\overline{\text{CAS}}$信号の発生

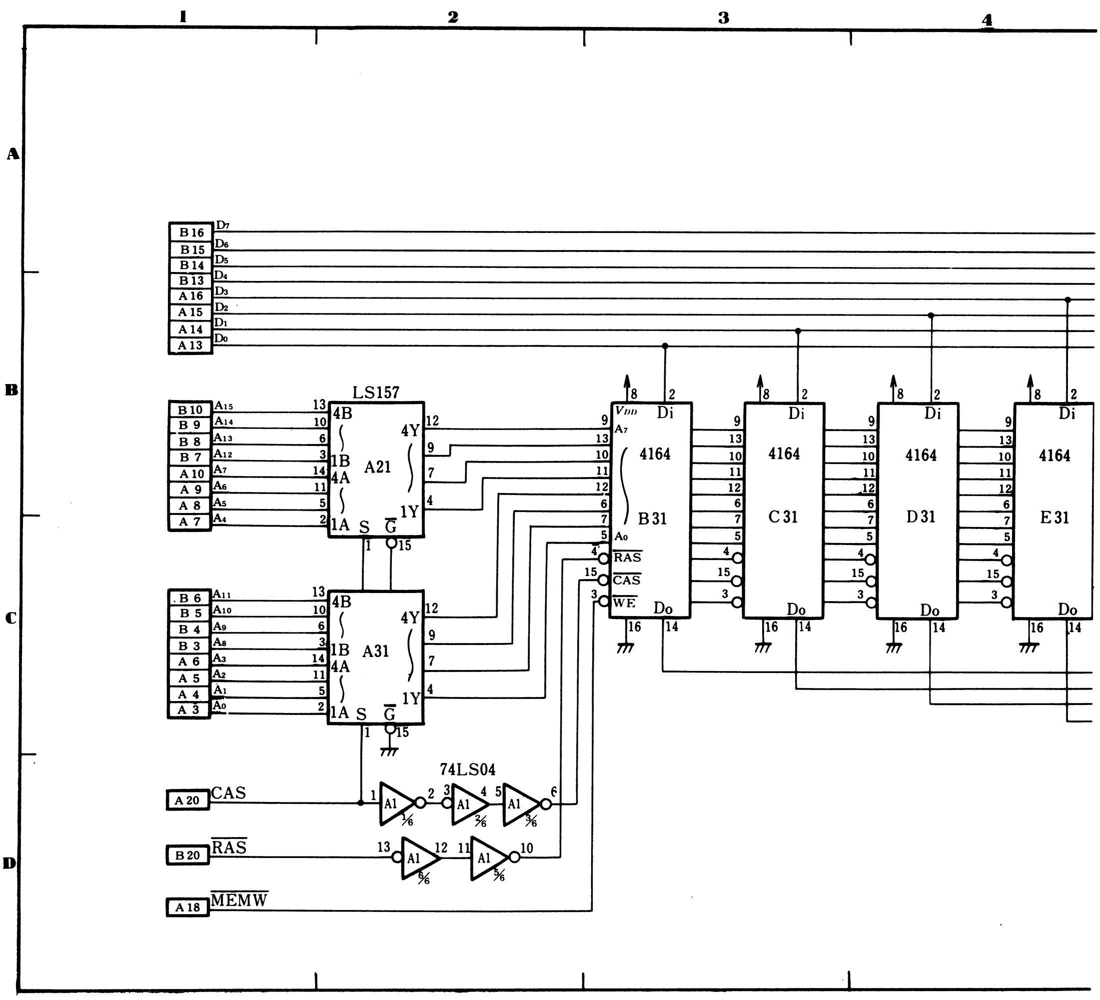


図4.36　64Kバイト

シュは$\overline{\text{RAS}}$信号のみを与えることにより実行されます。これをRASオンリーリフレッシュといい，次のようにリフレッシュ動作が行なわれます。

リフレッシュ期間中（$\overline{\text{RFSH}}$が“Low”）は，**図4.40**に示すようにCPUからアドレスバスA_0～A_7にリフレッシュアドレスと$\overline{\text{MREQ}}$信号が出力されます。そのため，$\overline{\text{RFSH}}$が“Low”のときは74 LS 74（図4.1の**3E**）の$\overline{\text{PReset}}$状態となります。

すなわち，CAS信号が“Low”となり，アドレス選択回路が行アドレス選択となり，アドレス

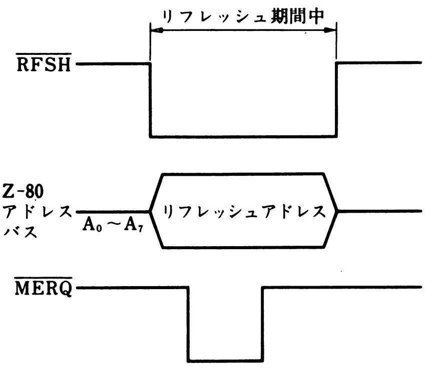
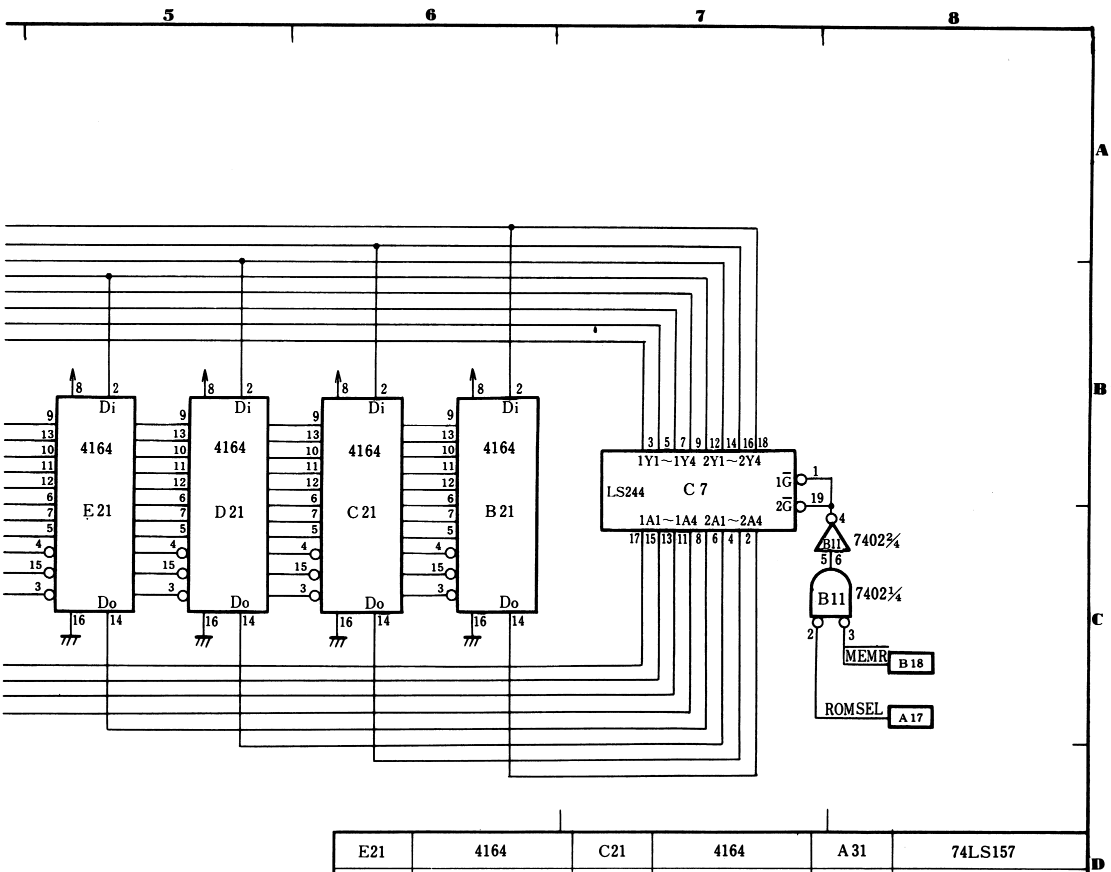


図4.40　リフレッシュ動作中のZ-80CPUの動作信号



		E21	4164	C21	4164	A31	74LS157
		D31	4164	B31	4164	A21	74LS157
		D21	4164	B21	4164	B11	7402
E31	4164	C31	4164	C 7	74LS244	A11	74LS04

メモリ基板の配線図

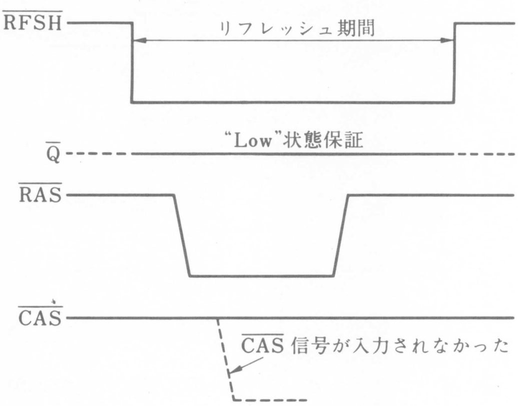


図4.41　$\overline{RAS}$オンリーリフレッシュの動作

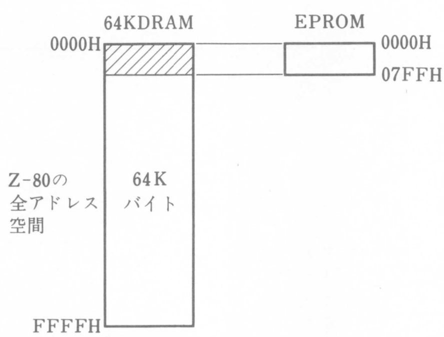


図4.42　EPROMのアドレス空間と競合するDRAM領域

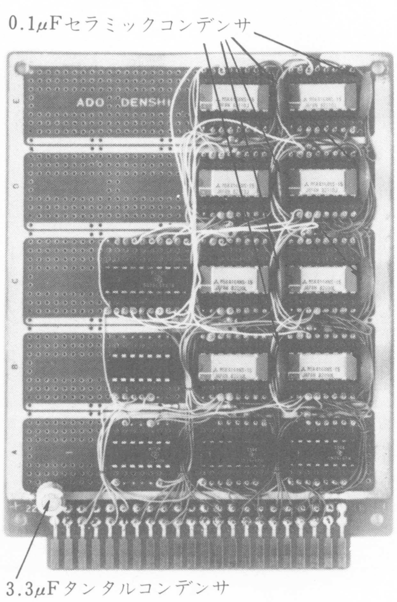


図4.43　作製したDRAM基板

バスの下位8ビット（リフレッシュアドレス）が出力されます。そして，図4.40の$\overline{MREQ}$が発生されることにより，$\overline{RAS}$信号のクロックがDRAMに加えられます。

そのとき，**図4.41**に示すように，$\overline{RAS}$クロックが発生（"High"から"Low"そして"High"に上昇）し終わっても，列アドレス選択信号$\overline{CAS}$は$\overline{RFSH}$信号により"High"になりつづけているため，"High"から"Low"への列指定動作が行なわれなかったことになります。

DRAMは，$\overline{RAS}$と$\overline{CAS}$との両方のクロックではじめて指定アドレスが定まります。そのため，$\overline{RAS}$のみのクロック入力のときは，メモリのデータのリード／ライトの作業は行なわれず，列アドレス番号に相当する図4.33の一列512素子が同時にリフレッシュされることになります。

■DRAMの選択

図4.36の図面中**7B**にあるTTL-LS 244（8入力のトライステートバッファ）は，DRAMのデータ選択用に利用されます。DRAMの領域は，**図4.42**に示すように64Kの全アドレス空間を占有するため，電源を入れてから最初のプログラムを動作させるのにEPROMが必要となり，このEPROMのアドレス空間を確保するため，EPROM動作中はDRAMを切り離しておかなければなりません。

この信号がメモリ配線基板端子番号A 17からの入力信号ROMSELECTです。そして，ROMSELECTが"High"の状態では，LS 244の1$\overline{G}$と2$\overline{G}$が"High"となり，DRAMのデータが遮断され，EPROMの出力データと競合しないことになります。

また，図4.36上，**8C**にある2入力AND回路の入力$\overline{MEMR}$は，DRAMのメモリ内容のリード時のみデータをCPU側に送信するために挿入してあります。

■実装上の注意

DRAMはコンデンサの電荷蓄積によって記憶するため，電荷注入時に過渡的に電流が流れ込みます。この注入時の電荷を補うために，各DRAMチップごとに周波数応答の比較良好な

セラミックコンデンサ(0.1 μF以上)を+5 V電源ピン付近に必ず挿入する必要があります。また，メモリ基板全体に対しては，5 V電源に3.3 μF以上のタンタルコンデンサを挿入しなければなりません。

このタンタルコンデンサは，プリント板端子番号A 22，B 22の5 V電源入力部にもっとも近い場所に配置します。**図4.43**にDRAMメモリ基板の配線ずみの写真を示します。

手作りコンピュータシステムの信頼性向上のために3.3 μFのタンタルコンデンサはこの DRAMプリント板と同様に，CPU基板やCRTC制御基板などの各プリント板に対しても必ず挿入した方がいいでしょう。

≪RS 232 Cインターフェース基板の作り方≫その2

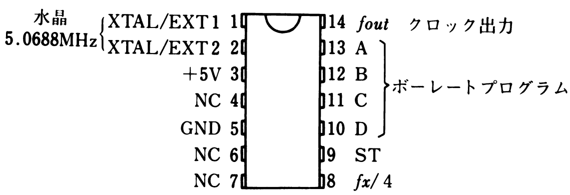
ボーレートジェネレータCOM 8146のピン配置を**付図C**に示します。このICは，端子ABCDをプログラムすることにより**付表B**に示すボーレート（通信送度ビット／秒）のクロックを作ることができます。ここで得られるクロックはボーレートの16倍になっています。

8251 AはプログラマブルなUSARTであり，初期設定コマンドかリセットにより，8251 Aはモード設定入力待ちになります。ここで**付図D**（次ページ参照）のモード命令フォーマットに従って，モード設定を行なっています。このモード設定は$C/\overline{D}$を“High”，$\overline{WR}$を“Low”，$\overline{CS}$を“Low”にして行ないます。

次に，$C/\overline{D}$を“Low”にしてデータを8251 Aに書き込むと，このデータはコマンドとして取り扱われます。**付図E**にコマンドのフォーマットを示します。

また，8251 Aは内部状態をCPUから読み出すことができます。ステータスは$C/\overline{D}$を“High”，$\overline{RD}$を“Low”にして読み出します。**付図F**にステータスのフォーマットを示します。

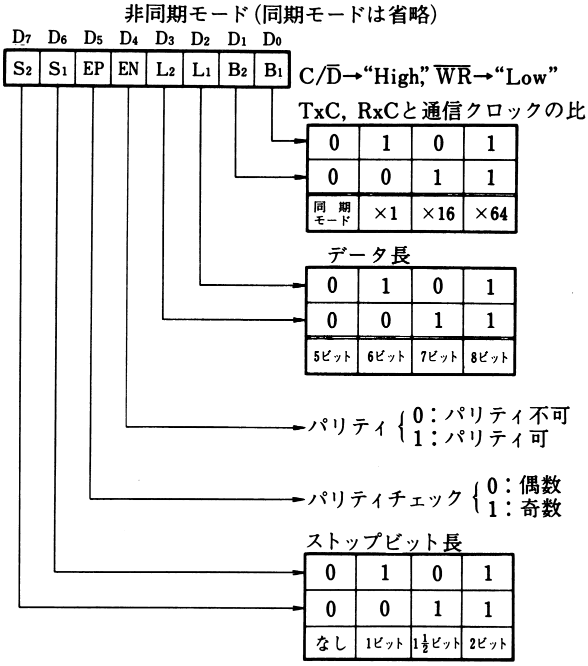
（76ページに続く）



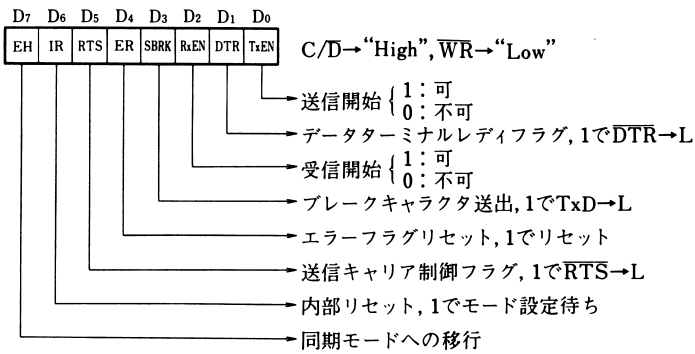
付図C　COM 8146のピン配置

付表B　COM 8146で得られるボーレート基準周波数5.0688 MHz

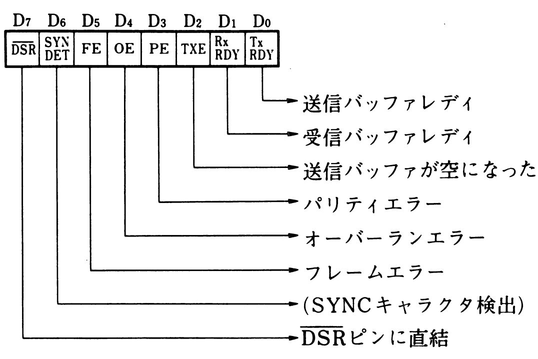
プログラム入力(DCBA)	ボーレート	クロックファクタ	周波数(kHz)	分周比	得られるボーレート	得られる周波数(kHz)	誤差
0000	50.00	×16	0.80000	6336	50.00	0.800000	0.0000%
0001	75.00	×16	1.20000	4224	75.00	1.200000	0.0000%
0010	110.00	×16	1.76000	2880	110.00	1.760000	0.0000%
0011	134.50	×16	2.15200	2355	134.52	2.152357	0.0166%
0100	150.00	×16	2.40000	2112	150.00	2.400000	0.0000%
0101	300.00	×16	4.80000	1056	300.00	4.800000	0.0000%
0110	600.00	×16	9.60000	528	600.00	9.600000	0.0000%
0111	1200.00	×16	19.20000	264	1200.00	19.200000	0.0000%
1000	1800.00	×16	28.80000	176	1800.00	28.800000	0.0000%
1001	2000.00	×16	32.00000	158	2005.06	32.081013	0.2532%
1010	2400.00	×16	38.40000	132	2400.00	38.400000	0.0000%
1011	3600.00	×16	57.60000	88	3600.00	57.600000	0.0000%
1100	4800.00	×16	76.80000	66	4800.00	76.800000	0.0000%
1101	7200.00	×16	115.20000	44	7200.00	115.200000	0.0000%
1110	9600.00	×16	153.60000	33	9600.00	153.600000	0.0000%
1111	19200.00	×16	307.20000	16	19800.00	316.800000	3.1250%



付図D　モード命令フォーマット



付図E　コマンドフォーマット



付図F　ステータス

第1部第5章

マイコン本体の作り方

5

第4章では，手作りコンピュータの基本回路の基板作製方法について述べました。ここでは，前章で作製した各基板を納めるマイコン本体の作り方と，配線について述べます。

5.1 本体の作り方

作製しているマイコンは，基板が最低7枚必要となります。そこで，図5.1に示すような7枚以上の基板が収納できるケースが市販されていますので,*それを入手して下さい。

ここでは，既成品を買わないで，約40×20×16 cmの箱形本体ケースを自作する手順について述べます。市販のケースが入手可能な場合は，この項はとばして，5.1.2項の“部品の取り付け”へ進んで下さい。

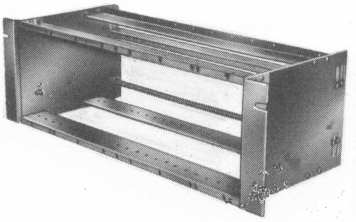


図5.1　市販されているケースの例（アイデアル RF-150カードファイル＝摂津金属工業㈱製）

5.1.1　本体の組立て手順

手作りコンピュータの本体として，約40 cm（幅）×20 cm（奥行き）×16 cm（高）の箱形のものを作製します。それでは，必要な材料と組立て手順について述べます。

* KEL㈱や㈱鈴蘭堂からも同様のものが市販されています。

■必要とする材料

工作がしやすく，入手しやすいアルミ材を用います。厚さ2 mmのL形アングルを用いてフレームを，また厚さ2 mmのアルミ板を用いて側面とします。

用意するものは，**表5.1**に示すように，厚さ2 mmのアルミで，30 mm×30 mmL形アングルの長さ120 cm以上のものを1本と，厚さ2 mmのアルミで15 mm×30 mmL形アングルの長さ140 cm以上のものを1本必要とします。これらのL形アングルを**図5.2(a)，(b)**に示す寸法に切断します。

側面用としては，厚さ2 mm*のアルミ板を40 cm×88 cmだけ用意して下さい。これを図5.2（c）に示すように切断します。

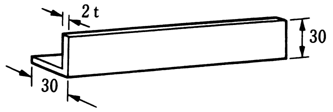
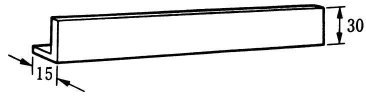
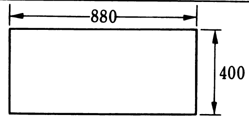
以上が本体を構成する基本材料です。

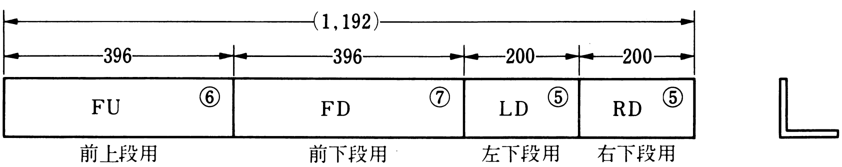
■本体の作り方

図5.2のように切断したアルミアングルとアルミ板に，**付録9**に示す図面に従って，ドリル

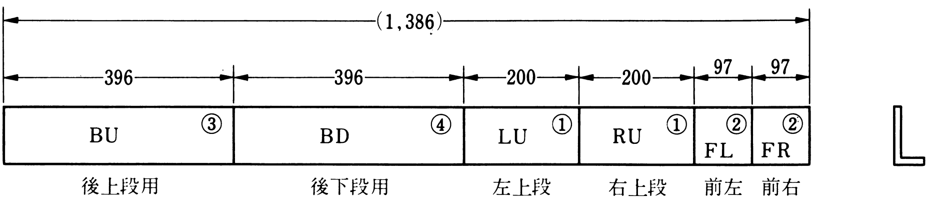
* 2 mmのアルミ板がもっとも適していますが，1.5 mmでも可能です。

表5.1　準備するアルミ材

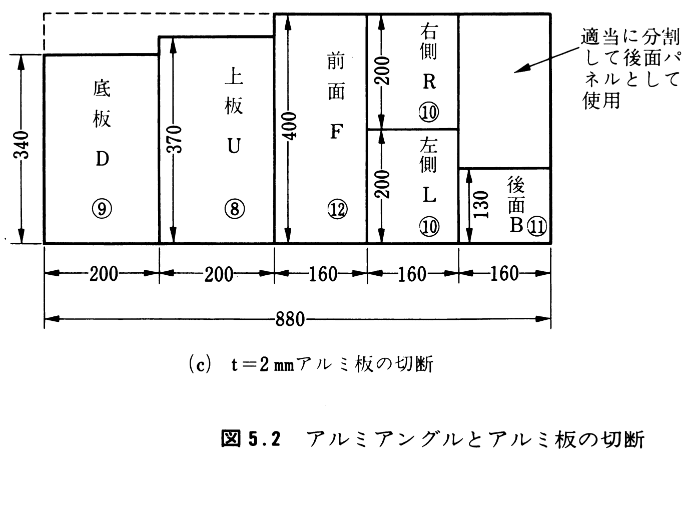
アルミL型アングル		長さ120cm以上
		長さ140cm以上
2㎜厚アルミ板		1 枚



(a)　L30×30アングルの切断



(b)　L30×15アングルの切断



(c)　t＝2㎜アルミ板の切断

図5.2　アルミアングルとアルミ板の切断

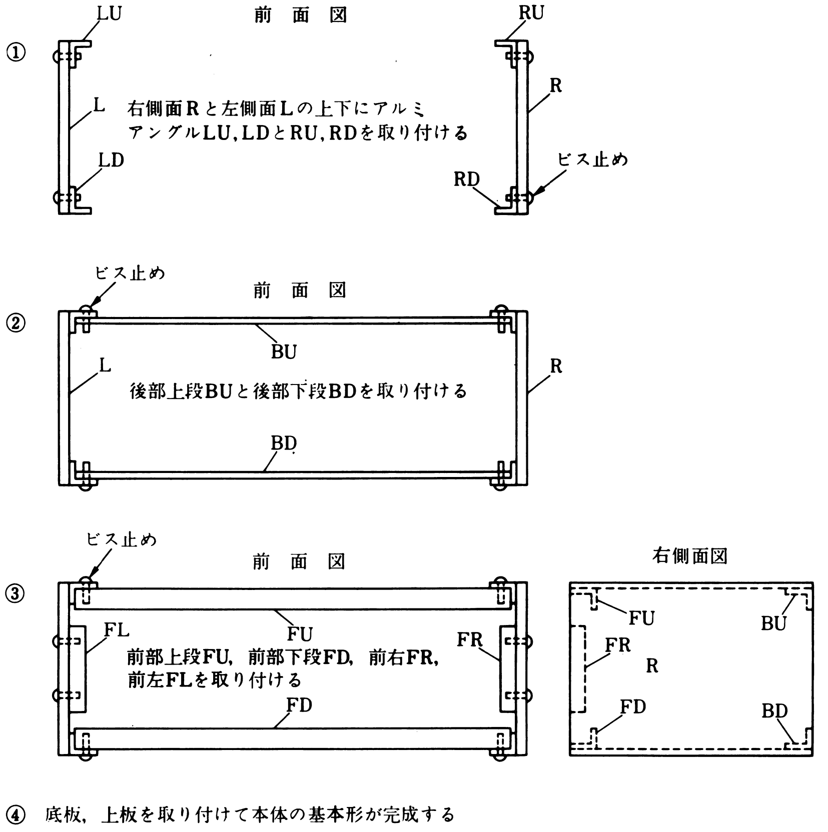


図 5.3 本体の組立て順序

穴をあけます。3㎜ネジでネジ止めする穴は2.4㎜の下穴に3㎜のタップを立てて下さい。各材料を，**図 5.3** に示す①～④の順序で組み立てて下さい。これで電源スイッチやパイロットランプを取り付ける前面パネルと，ヒューズなどを取り付ける後面パネルを除いて本体が完成します。

5.1.2 部品の取付け

■電気部品の取付け

前面パネルには，100 Vの電源をON，OFFするメインスイッチと，その電源がONされたときの状態を示すパイロットランプおよびZ-80 CPUリセット用のプッシュスイッチ（押したときに導通状態となり，離せば切断状態となるスイッチ）を取り付けて下さい。

後部パネル（図 5.2 (c) の“B”）には，ヒューズ，100 Vの電源コードを通すブッシュを取り付けます。

■基板ソケットの取付け

本体前面に取り付けたアルミアングルFUとFDに，各配線基板を挿入する44ピンソケット（以下，ピンはPと略します）を8個，22 Pソケットを1個取り付けます。

このシステムでは，44 Pコネクタは8個のうち6個のみを使います。残りの2個は，将来，入出力装置を追加するときに使用するソケット用です。

各ソケットの配置は，**図 5.4** に示すようにして左側寄りに取り付けて下さい。右側の空間は，スイッチングレギュレータなどの電源部品を取り付けるためのスペースです。

44 Pのコネクタは，上部が番号1，下部が番号44端子となるように揃え，同様に22 Pのコネク

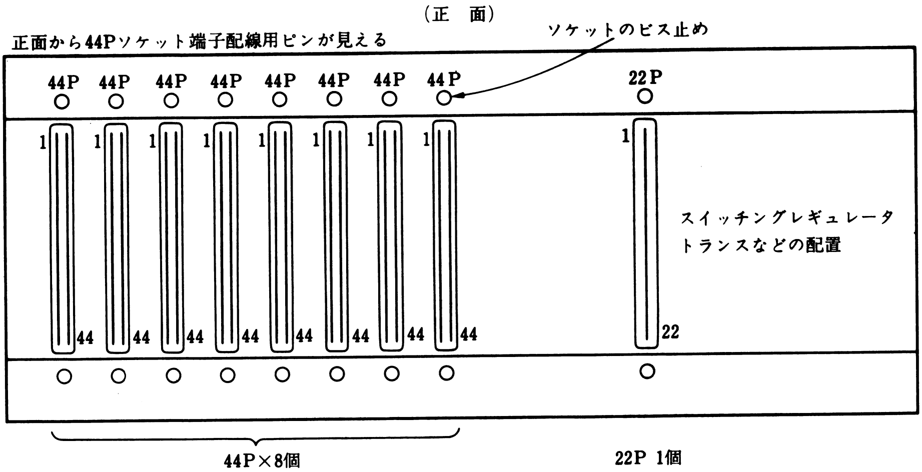


図 5.4　基板ソケットの取付け

タは，下部が番号22となるように取り付けます。こうすると，正面側にソケットの配線用の端子が取り付けられ，各配線基板は裏側から挿入することになります。このようにソケットを取り付けると，基板の部品がつく側は，向かって左側の面となります。

5.2 本体の配線

第5章　マイコン本体の作り方

電気部品が取り付けられたコンピュータ本体の配線方法について述べます。

5.2.1　直流電源の作製方法

この手作りコンピュータは，主としてTTLコンパチブルとなっているため，直流５Ｖで動作します。しかし，付属の装置を動作させるのに，**表 5.2** に示す各種電圧の電源を必要とします。

ここでは，標準単密フロッピーディスクYD-74Ｃ（ワイ・イー・データ社製）を１台取り付けることを前提として必要とする電圧を定めています。したがって，購入するフロッピーディスクなどの付属装置のカタログを参照して，各自必要とする電圧とその電流容量を算出して下さい。

■スイッチング電源

このシステムでは，５Ｖ５Ａの電源が最低必要です。５Ｖ５Ａの電源を作ることはかなり困難な作業ですので，現在安価に市販されているスイッチング電源を購入して下さい。

スイッチング電源には，各種インターフェースに必要とする電圧値を考慮したものがあり，５Ｖ５Ａ以外の電圧出力端子が取り付けられている製品が市販されています。表5.2の合計の欄を考慮して，さらに追加する電源がもっとも少なくなるようなタイプを購入して下さい。

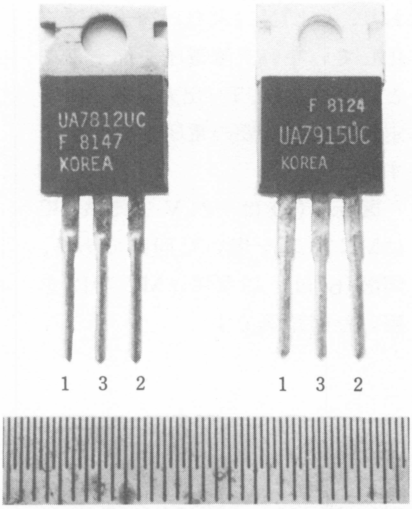
■ 3端子レギュレータ

スイッチング電源では得られなかった電圧を発生させるには，トランスとダイオードおよび３端子レギュレータを用いて直流安定化電源を作製します。

３端子レギュレータには，**表 5.3** に示すように，正電圧発生型の78シリーズと負電圧発生型の79シリーズがあり，電流容量に従って各種の

表 5.2 必要とする電源

装置名	仕様	備考
各種基板 TTL 動作用	＋5 A　4 A	
キーボード	－12 V　50 mA	－12 V を必要としないキーボードを購入した場合は不要
フロッピーディスク 本体 YD-74C	＋5 V　1 A －12 V　100 mA ＋24 V　1.5 A	購入するフロッピーディスクのカタログ参照のこと
フロッピーディスク インターフェース IC　1771	＋12 V　50 mA －5 V　50 mA	1771以外の FDC IC を使用する場合はカタログ参照のこと
合計	＋5 V　5 A ＋24 V　1.5 A ＋12 V　50 mA －5 V　50 mA －12 V　100 mA	

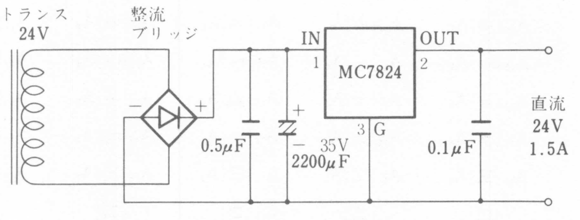


ピン 1	入力	ピン 1	GND
ピン 2	出力	ピン 2	出力
ピン 3	GND	ピン 3	入力

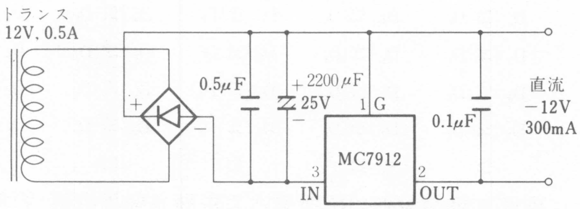
図 5.5 3端子レギュレータ 78××と 79×× のピン配置

表 5.3 3端子レギュレータの型

	型名	仕様
正電圧	78L××	0.1A
	78M××	0.5A
	78××	1.5A
	78H××	5A
負電圧	79L××	－0.1A
	79M××	－0.5A
	79××	－1.5A



(a) 直流＋24 V電源



(b) 直流－12 V電源

図 5.6 3端子レギュレータの使い方

タイプが市販されています。

この3端子レギュレータは，**図5.5**に示すようなピン配置をしており，GとIN（入力）端子に安定化していない直流電圧を加えることにより，OUT（出力）端子に安定化された所要の電圧が発生します。

図5.6(a)は＋24 V（1.5 A）用にMC 7824を用いた回路であり，同図**(b)**は－12 V用にMC 7912を用いた回路例です。

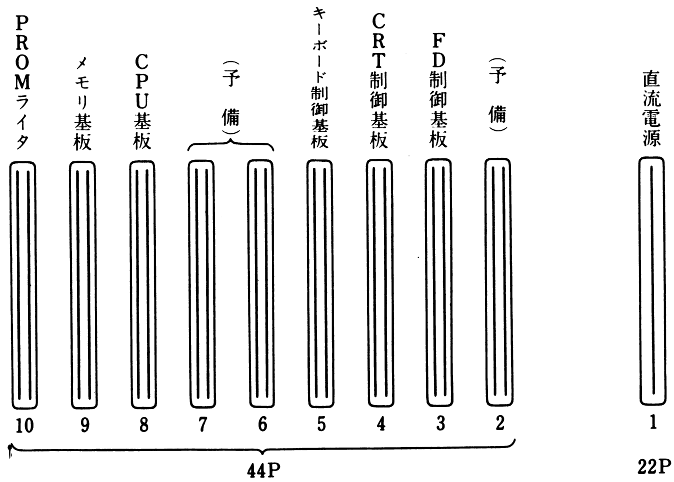


図5.7　配線基板の配置

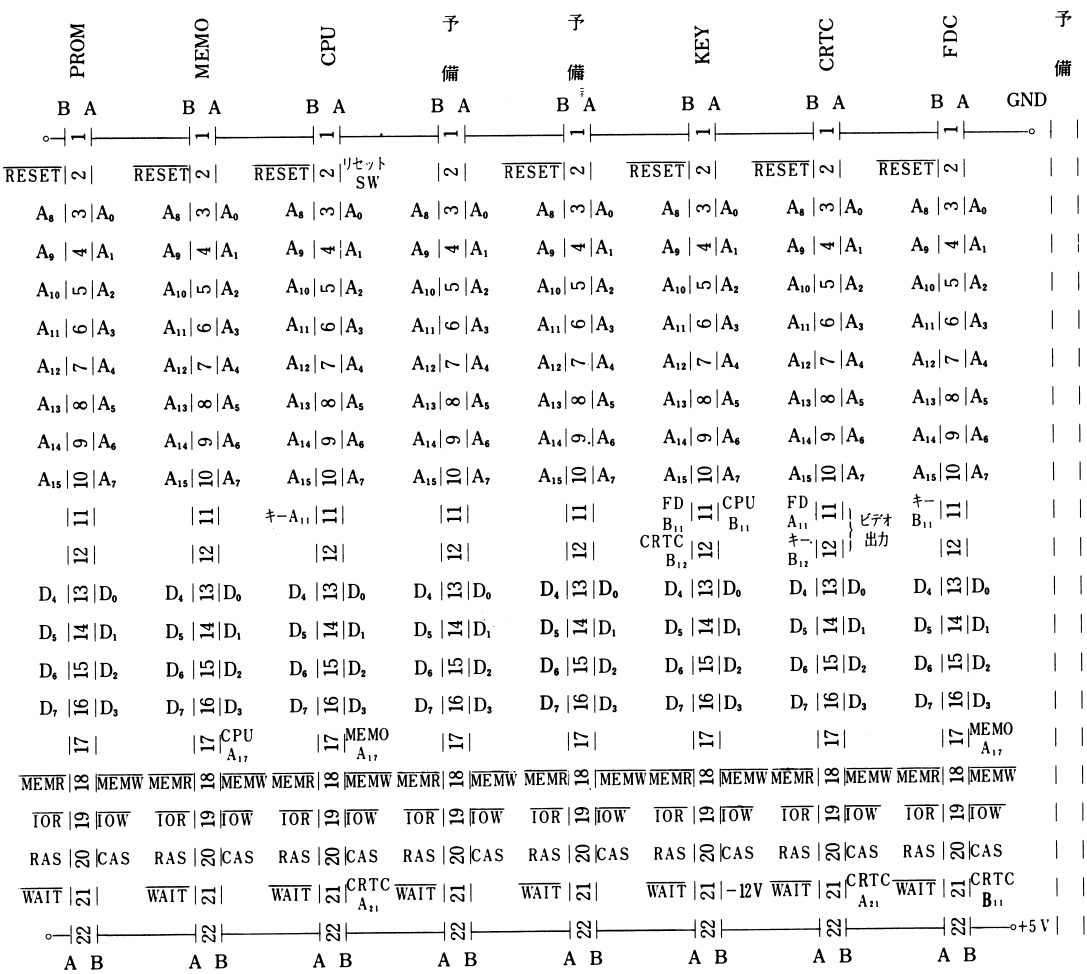


図5.8　基板ソケット部の配線図

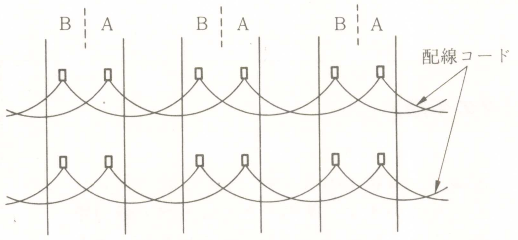
5.2.2　基板ソケットの配線

本体には，**図 5.7** に示すように合計 10 枚の配線基板が挿入されます。各配線基板の順序を図のように定め，これらの基板ソケット端子の配線方法について示します。

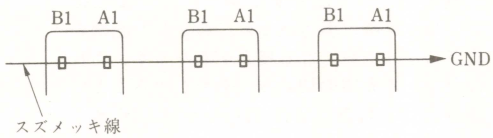
■ 22 Pソケットの配線

図 5.7 の右側には 22 Pのソケットをつけます。この基板は直流電源作製用で，前項で述べた 3 端子レギュレータを用いた各種直流安定化電源回路を搭載した基板を挿入します。

この端子の配線順序はとくに定めませんので，各自必要に応じて 22 Pの端子の配線を行ない，他の基板へ電圧を供給するようにして下さい。



(a)　全ソケット共通端子の配線



(b)　GNDと5Vの配線方法

図 5.9　基板ソケットの配線方法

■ 44 Pのソケットの配線

9 個の 44 Pソケットの配置図を図 5.7 に示しました。使用するソケットは 44 Pですので，配線基板の端子番号に合わせ，A 1～A 22 およびB 1～B 22 に端子ピン番号を定めます。

配線には 2 種類あり，全部のソケット端子に共通な配線と，各配線基板のみに固有の配線です。共通な配線は，**図 5.8** に示すように $\overline{\text{RESET}}$ (A 2)，A_0 (A 3) ～A_{15} (B 10)，D_0 (A 13) ～D_7 (B 16) および $\overline{\text{MEMW}}$ (A 18)，$\overline{\text{MEMR}}$ (B 18)，$\overline{\text{IOW}}$ (A 19)，$\overline{\text{IOR}}$ (B 19)，$\overline{\text{WAIT}}$ (B 21) です。

これらは図面中の端子がすべて共通配線ですから，**図 5.9 (a)** に示すように配線します。

また，すべてのソケットのA 1，B 1 端子はGNDであり，A 22，B 22 端子は 5 Vなので，図

表 5.4　各基板に固有な配線

基板名	ピン番号	行先基板名	ピン番号	信号の内容
FDC 基板	A 21	CRTC	B 11	4 MHz クロック
	B 11	キーボード基板	B 11	WAIT 信号入力
CRTC 基板	A 11	―	―	GND } グリーンディスプレイへ
	A 12	―	―	ビデオ出力 } グリーンディスプレイへ
	B 11	FDC 基板	A 21	4 MHz クロック
	B 12	キーボード基板	B 12	ブランキング信号入力
キーボード基板	A 11	CPU 基板	B 11	バンクセレクト信号
	B 11	FDC 基板	B 11	WAIT 信号入力
	B 12	CRTC 基板	B 12	ブランキング信号
CPU 基板	A 2	―	―	外付けリセットスイッチ
	B 11	キーボード基板	A 11	バンクセレクト信号
	A 21	CRTC	A 21	クロック ϕ
	A 17	MEMO 基板	A 17	ROM セレクト信号
MEMO 基板	A 17	CPU 基板	A 17	ROM セレクト信号

5.9（b）に示すように太いスズメッキ線で配線します。その他の基板に固有な配線は，**表5.4**に示すように数本ありますので，図面に従って配線して下さい。

■各基板に固有な配線

全基板に共通な配線以外の固有な配線を表5.4に示しましたが，この表の見方としては，例えばFDC基板に関してはA21とCRTCのB11，およびB11とキーボード基板のB11を配線するということを示しています。

これらの配線を行なうことにより，各基板の位置が決定されます。配線を行なった後は，間違ってほかの基板を挿入しないように，ソケットの近くに油性のサインペンで基板名を書いておくと便利です。これで本体の配線はすべて完了したことになります。

《正しいラッピングの手順》

① ラッピング線の先から25～30㎜分の被覆を取り除く。

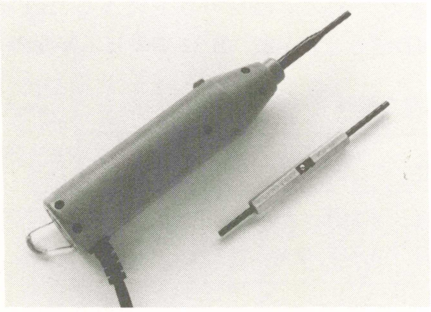
② ラッパの先に線を挿入する。そのとき必ず被覆部分も3～4㎜挿入します。

③ ラッピングピンをラッパの中心の穴へ挿入し，ラッパを時計回りにまわす。その際，軽く引き上げるようにします。

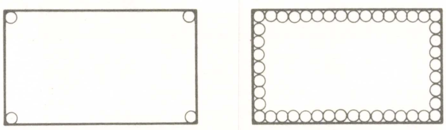
ただし，無理に引き上げると，線の間隔があいてしまうので注意が必要です。また，ビニル被覆部分が2回程度まかれていなければなりません。

④ 片側のラッピングが終わったら，その線の配線すべき相手のラッピングピンにラッピングを行ないます。そのとき，なるべく最短距離になるように，すなわち線の長さができるだけ短くなるように配線の経路を選び，①～③に従ってラッピングを行ないます。

⑤ 電動ラッパを使用する場合は，しばしばラッピング線が引き込まれ，設定した配線の長さにより，短くなる場合があります。これはラッパの先端の切込みに線をかけることで防げます。**付図5.1**に電動ラッパと手巻きラッパを示します。



付図5.1　電動ラッパと手巻きラッパ



(a) 四隅にドリル穴をあける　(b) 四辺に沿ってドリル穴をあける

付図5.2　四角形の穴のあけ方

《取付け穴のあけ方》

電源スイッチ，パイロットランプなどを取り付けるには，購入した部品に合った直径の穴をあけなければなりません。その場合，指定した穴があくようなドリルが利用できればいいですが，利用できない場合は，小さな穴を大きくする工具にリーマがあります。

このリーマの先をドリル穴に差し込み，リーマを押し付けながら回転させると，少しずつ穴が大きくなります。日曜大工の店でもリーマという名で入手できます。

一方，四角形の穴を効率良くかつ美しくあけるには，四角形の四隅にドリル穴をあけることでも可能です。糸ノコをもっていない人は**付図5.2**に示すように，四角形に沿って，多くの小さな穴をドリルであけ，穴と穴の間を小形ヤスリなどで切断して切り落とし，断面を平ヤスリで仕上げると美しくできます。

第1部第6章

PROMライタの作り方 6

ここでは，PROMライタをもっていない読者のために，手作りマイコンシステムで動作するPROMライタの作り方について述べます。すでにもっている読者は，本章をとばして第7章へ進んで下さい。

マイクロコンピュータを起動させるためには，初期動作のプログラムが必要であり，また，プログラムを記憶しておくためにはメモリが必要です。RAMメモリは電源を切ることによって内容がすべて失われてしまいます。したがって，電源を切っても内容の消えることのない初期プログラムをメモリ上に準備しておかなければなりません。そのために，電源を切っても内容が消えないROMがあります。

6.1 UVEPROMと消去法

ROMには，大別してマスクROMとフィールドPROM（Field Programmable ROM）とがあります。マスクROMは内容固定のROMであり，フィールドPROMはユーザが電気的にプログラムの書込みが可能なROMです。フィールドPROMのなかで，ユーザによりプログラム内容を何回でも自由に消去，書込みができるのがEPROM（Erasable Programmable ROM）です。

手作りコンピュータシステムでは，一般的に広く利用されているUVEPROM（Ultra-Violet Erasable ROM）を用います。これは紫外線によりプログラム内容の消去ができるものです。以下，UVEPROMの消去方法について述べます。

■UVEPROM

現在，各種のUVEPROMが市販されていますが，安定した品質で，かつ安価に出まわっているのが，2Kバイトの記憶容量をもつ2716と，4Kバイトの2732です。ここでは，4Kバイト分のROM領域を必要としないので，安価な2716を利用することにします。2716は，書込みや消去がきわめて簡単に行なえるため，多くのマイコンシステムで使われています。

■UVEPROMの消去法

EPROMへの書込み方法については，次節で述べますが，ここでは書き込まれているプログラムの消去方法について述べます。

図6.1は，UVEPROM 2716の外観を示したものですが，中央に窓があり，中のパターンを観察できるのが特徴です。

この窓は，パターンに対して外部から紫外線をあてることが目的で，紫外線により書き込まれているプログラムの消去が行なえます。なお，

消去を行なうと，すべてのビットが“High”となり，消去の確認が行なえます。

消去方法は，ROMイレーザを用いるか，または電気店で売っている15 Wの蛍光灯スタンドと殺菌ランプ（例えば東芝殺菌ランプGL 15）を利用します。

この殺菌ランプで消去するには，**図6.2**のように，2716を殺菌ランプに近づけ，消去窓に約20分間紫外線を照射することにより，消去が完了します。

殺菌灯を使用するときは，長時間直接紫外線を目に入れたり紫外線が照射されているものを見ていると，目を傷めることがありますので，注意しましょう。殺菌灯は押入れやロッカーなどの中に入れ，紫外線がもれない状態で使用して下さい。

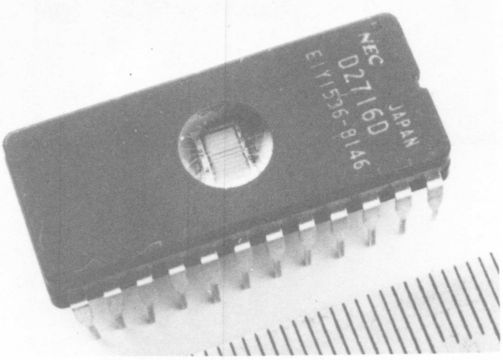


図6.1　UVEPROM 2716の外観

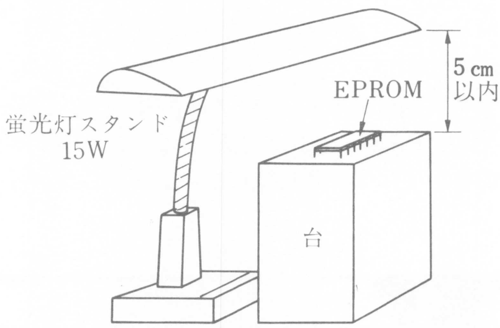


図6.2　殺菌灯による消去方法

第6章　PROMライタの作り方

6.2 PROMライタ回路について

ここでは，EPROMおよび試作するPROMライタ回路の設計方法について紹介します。

6.2.1　2K ROMライタ回路の構成

■ **2716の書込み**

2KバイトのUVEPROM 2716は，**図6.3**に示すピン接続の構成で，入出力すべてがTTLコンパチブルとなっています。また，1バイトだけの書込みも可能となっています。

2716に対する書込み動作を**図6.4**に示します。この2716のアドレスA_0〜A_{10}に書き込むべき番地を指定し，D_0〜D_7に書き込むべきデータ値を設定した後，$\overline{\text{CS}}$およびPD／PGMに対して図6.4の仕様に従ったパルス波形を加えれば，書込みが終了します。

このとき，$\overline{\text{CS}}$（チップセレクト）は，“Low”レベルで2716のREAD状態，また“High”でWRITE状態です。したがって，書込みに先立ち，$\overline{\text{CS}}$を“Low”から“High”に設定し，その後で書込みタイミングパルスPD／PGMを加えることになります。このように，PROMライタは，$\overline{\text{CS}}$とPD／PGMの2つのパルス波形の発生動作が主になります。

■ **PROMライタ回路**

PROM 2716，2732兼用PROMライタを製作します。これは，手動ライタと手作りコンピュー

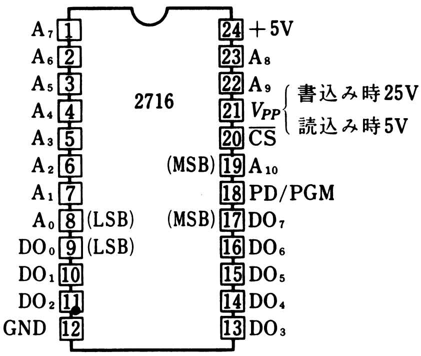


図 6.3　2716 のピン配置

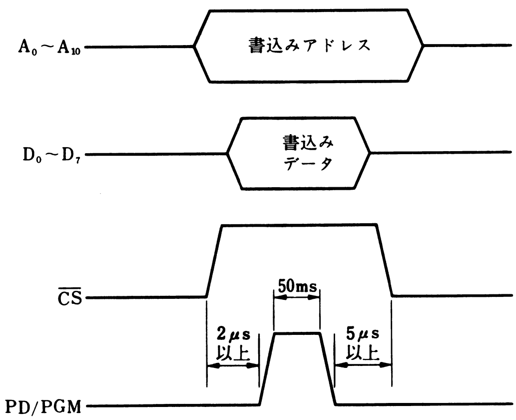
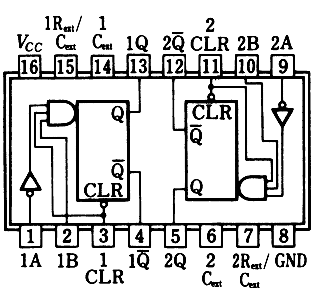


図 6.4　書込み動作



INPUTS			OUTPUTS	
CLEAR	A	B	Q	$\bar{Q}$
L	×	×	L	H
×	H	×	L	H
×	×	L	L	H
H	L	↑	⊓	⊔
H	↓	H	⊓	⊔
↑	L	H	⊓	⊔

図 6.6　TTL 74 LS 123 のピン配置と動作

タを用いた自動ライタの兼用使用ができるものです。

巻末折込みページの**図 6.5** に，試作する PROMライタ回路の配線図を示します。図中の **2C**，**2D** の単安定フリップフロップ 74 LS 123 を用いて，2 つのパルス$\overline{\mathrm{CS}}$とPD／PGMを発生させます。

74 LS 123 は 1 パッケージに 2 つの回路が組み込まれています。**図 6.6** はピン配置を示しており，A端子入力の立ち下がりまたはB端子入力の立ち上がりのタイミングで，CとRにより定まる時間，すなわち

$$t_w = 0.45\ R_T \cdot C_T$$

のパルスが発生します。

ここでは**図 6.7** に示すように，3 つの単安定回路を利用して$\overline{\mathrm{CS}}$とPD／PGMを発生してます。とくに，PD／PGMは，パルス幅が 45 ms～55 msと定められていますので，CR回路の抵抗値をボリュームによって変えて，正しく **50 msのパルス幅信号**が得られるように調整して下さい。

また同様にして，$\overline{\mathrm{CS}}$信号のパルス幅もボリュームを変えて，PD／PGM信号が終了しても最低 5 μsは“High”状態を保つように多少長目に設定して下さい。

■DC-DCコンバータ

2716 は書込み動作の際，図 6.3 のピン配置図における 21 番ピンV$_{PP}$に対し，+25 Vを印加しておかなければなりません。ここでは，TI社製のDC-DCコンバータ用IC・TL 497 ACNを用いて，5Vから昇圧させて電圧を得ることにします。

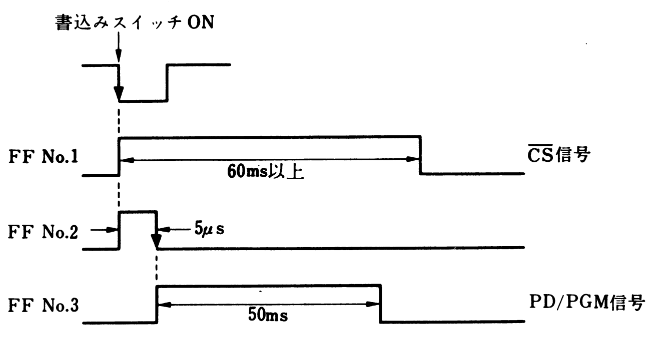


図 6.7　書込み信号パルスの発生

図6.5の，PROMライタ配線図面の**7C**に配置されているTL 497 ACNがDC-DCコンバータ回路部分です。

出力電圧は，10 kΩのボリュームによって調整されますので，電圧計を見ながら25 Vが出力されるように設定します。

なお，注意しなければならないのは，V_{PP}端子に26 V以上の電圧は絶対に加えないことです。26 V以上だとEPROMが壊れますので，DC-DCコンバータの電圧設定には細心の注意を払うことが必要です。

13番ピンと10番ピンの間には150 μHのコイルを挿入します。150 μHのLは，電気パーツ屋でDC-DCコンバータ497用のコイルと指定して入手して下さい。

25 V電圧は，書込み時のみに印加するため，通常は25 Vを印加することのないように，+25 V電源用スイッチを取り付けます。図6.5の**7C**に配置したSW 2が25 V電源用スイッチです。このスイッチには，ONにした瞬間の過渡電流を補うため，スイッチの**5 V入力側に**必ず6 V 100 μF以上の電解コンデンサを入れておきます。また，本基板の+5 VのみをON/OFFするスイッチ（図面内**2B**）も取り付けて下さい。

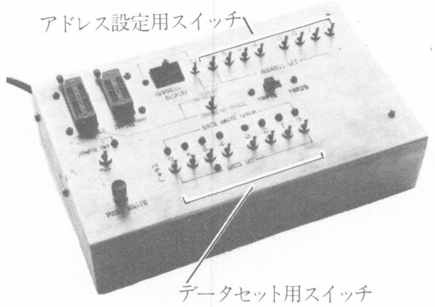
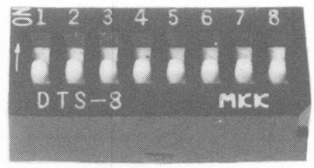


図6.8　スイッチを用いたEPROMライタ



(a) スライド式（使用不可）



(b) シーソー式（使用可）

図6.9　DIPスイッチ

■使用するDIPスイッチの仕様

図6.5の配線図面の**7C**にはアドレス設定用に11回路分のDIPスイッチを，また同図面中**4B**にはデータ設定用の8回路分のDIPスイッチを使用します。

これは，手動による書込みの際に使うスイッ

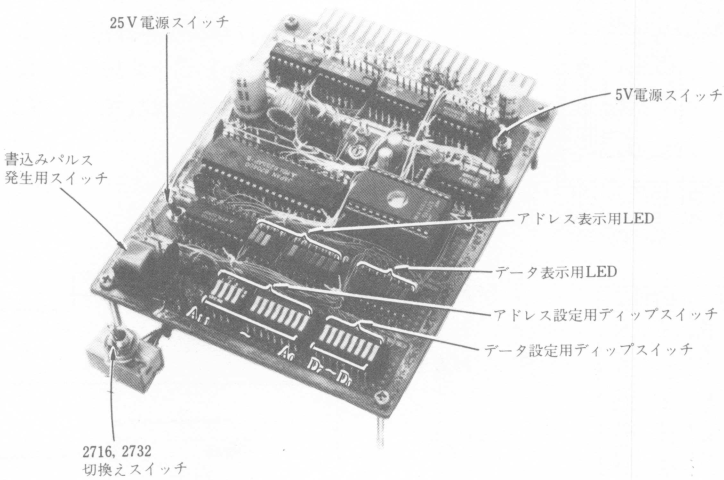


図6.10　EPROMライタの配線基板

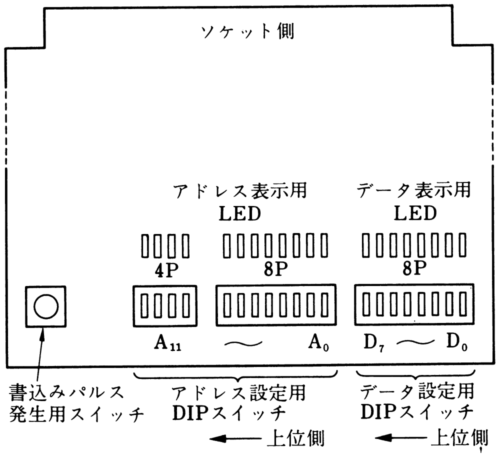


図 6.11 DIPスイッチの配置図

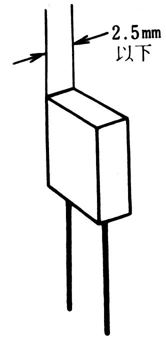


図 6.12 幅の狭い角型LED（幅 2.5 mm以下）

チで，ふつう，**図 6.8** に示すようなON，OFFスイッチが使用されています。しかし，ここでは使用するスイッチの数が多く，高価となり，外付けボックスが必要となりますので，代わりに安価なDIPスイッチを使用します。

このシステムでは，主としてCPUを使用した自動書込みを行ないますので，多少不便でも手動書込みの際はDIPスイッチで十分と考えられます。ただし，使用するときの使いやすさを考慮して，**図 6.9 (a)** に示すスライド式よりも図 6.9(b)に示すようなピアノ型やシーソー型DIPスイッチを使用して下さい。

■DIPスイッチの配置

図 6.10 に配線を終了した基板の写真を示します。このようにDIPスイッチは基板上の**E**の列に配置します。配置順序は，**図 6.11** に示すように，左側から順に 4 Pと 8 Pを並べて 12 本のアドレス設定用スイッチを固定し，また右側に 8 Pのデータ設定用のDIPスイッチを固定して下さい。

また，基板**E**の列の左端には，手動による書込みタイミングパルス発生用の押ボタンスイッチを取り付けて下さい。このスイッチは，図 6.5 の配線図画中では **1D** に示すPUSH-SWに相当します。

■表示用LEDの配置

図 6.11 に示したように，アドレス設定用とデータ設定用のDIPスイッチとを対応させて，各対応ビットを表示するLEDを配置します。このとき，DIPスイッチの間隔と等しい間隔にLEDを配置するのが望ましく，できれば**図 6.12** に示すような幅 2.5 mm以内の角型LEDを使用するほうがいいでしょう。LEDを新しく購入するのであれば，丸型のものではなく，2.5 mmピッチで配置できるような小型のLEDを入手するようにするといいと思います。

DIPスイッチは安価なので，不確実な動作のものもあり，一見ONになっているように見えても接続されていない場合も出てきます。ON状態の確認のためにも表示用LEDは必ず取り付けて下さい。

6.2.2 手動によるROMへの書込み

次に，手動によるROMへの書込み手順について述べます。

■手動と自動の切換え

図 6．5 に示したEPROMライタ回路は，手動書込みだけでなく，コンピュータが完成すれば，Z-80 CPUの動作によって，自動でプログラムを書き込んだり，PROMの内容を読み出したりすることができるように設計されています。

手動と自動の切換えは 8255 Aのリセット信号を用いて行ないます。また，手動の際は 8255 AにRESETパルスを印加し，**RESET状態**，すなわち全ポートを入力モードに設定します。

RESETパルスは，外付けのリセットスイッ

チを押すことにより生じた$\overline{\text{RESET}}$信号を用いることにします。

しかし，$\overline{\text{RESET}}$信号が印加された後，CPUのプログラム動作により8255 Aの各ポートに信号が出力されると，手動の設定値通りに書き込まれなくなります。この危険を回避するために，**8255 Aを強制的にRESET状態にする手動スイッチを取り付ける**ことが得策です。手動設定スイッチの構成を図6.5の**6A**の囲み内に示しておきましたので，必要と思われる読者は付け加えて下さい。

■ 25 V電源スイッチの取扱い

EPROM書込み動作においては，21番ピンに＋25 Vを加えておかなければなりません。＋25 VをEPROMへ供給するスイッチが図6.5の**7C**にあるSW 2であり，このスイッチの取扱いはとくに慎重にしなければなりません。

すなわち25 Vを印加する際は，必ず**5 V電源が供給されていることを確認してから25 VスイッチをONにする**こと，また，動作終了やEPROMを取りはずす際には，必ず**25 VスイッチをOFFにしてから5 V電源をOFFにする**ことが重要です。

これについては，**図6.13**に順序を図示しておきましたので必ず守って下さい。

26 V以上の電圧を印加するとEPROMは壊れ

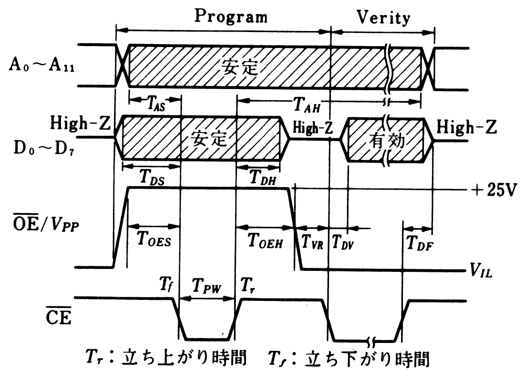
《4K　EPROM2732の書込み回路》

EPROMとして，4K容量の2732を利用する予定の読者は，EPROMライタは2732用にも利用できます。

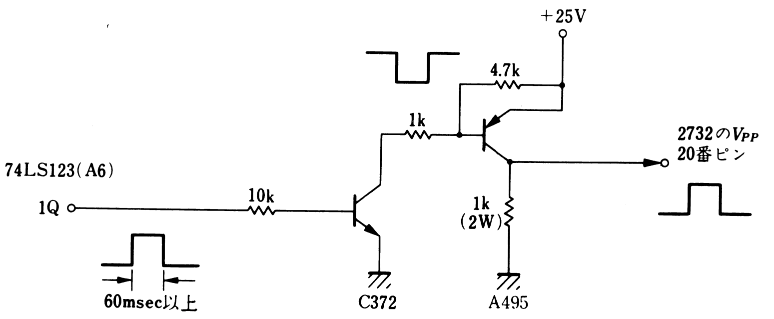
付図4.2に示したように，書込みモードではV_{PP}の位置が変わり，2716とのコンパチビリティはないので，切換えスイッチを用いて，以下のように変更しています。

21番ピンはA_{11}となるので，CポートのPC_3をアドレス信号として接続しています。また，20番ピンはV_{PP}となるので，スイッチを介して＋25 Vが印加できるようにしています。

このとき，書込みタイミングは，**付図6.1**に示すように＋25 Vのパルス波形を印加させなければならないので，**付図6.2**に示すような昇圧用のトランジスタ回路を用いています。また，トランジェントによる素子の破壊をさけるため，$\overline{\text{OE}}/V_{PP}$とGND間に0.1 μFのコンデンサを挿入します。



付図6.1　2716の書込みタイミング



付図6.2　50 ms幅の25 V_{PP}パルスの発生



図 6.13　電源スイッチの取扱い順序

ることがあります。はじめはDC電圧を**24.7V**程度と低目に設定して，書込みを行なって下さい。もしも24.7Vで書込みが行なえなかったら徐々に電圧を25Vまで上昇させます。

■手動PROMライタの動作テスト

EPROMライタ回路の配線が終了したら，18ページに示した拡張コネクタを介して，基板を図5.7の基板配置図の左端のソケットに挿入し，外部リセットスイッチを押して下さい。8255Aをリセット状態にして，次のように簡単な動作テストを行ないます。

はじめに，全部消去ずみのEPROM 2716をソケットに挿入し，5V電源だけをONにします。次に，任意のアドレスを設定し，8ビットのEPROMのデータ出力の8ビット分すべてが“High”になっているかどうかを確認して下さい。

図6.5の配線図に示すように，アドレス設定用とデータ設定用のDIPスイッチは，**導通時が“High”状態，断時が“Low”状態**となっています。また，LEDの表示は，**“High”状態で消灯，“Low”状態で点灯**となっていますので注意する必要があります。

以上のように，アドレス0000Hは，すべてのアドレス設定スイッチを導通状態にすることにより得られます。また，データがFFHであることは，データ表示用のLEDがすべて消灯していることにより確認できることになります。

■書込みテスト

次に+25V電源スイッチをON状態にして，データ設定用のDIPスイッチに任意のデータを設定し，書込みタイミング発生用の押しボタンスイッチを押して，瞬間だけONにして下さい。回路が正常に動作すれば，データ表示用のLEDが設定データと同じ内容を表示し，書込み動作が正しく行なわれたことが確認できます。

もし，正しく書込みが完了しなかった場合は，+25V電源が正しくEPROMの21番ピンに印加されているかどうか，また押ボタンスイッチにより正しく書込みパルスが発生しているかどうかを確認して下さい。そして，書込みが正しく終了したら，+25V電源スイッチを切り，任意のアドレスを設定すると，その内容データが正しくデータ表示LEDに示されるかどうかも確認して下さい。

《RS 232 Cインターフェース基板の作り方》その3

RS 232 Cインターフェースの入出力プログラムを，東京大学大型計算機センターのTSS端末として利用する場合を例にして説明します。TSSは，**付図G**のように音響カプラ(例えば，田村電気ACTAM-III，松下通信工業VC-246 B)と電話回線を通して，センターの大型計算機と接続します。

CP／Mマイコンを利用すれば，ソースプログラムの作成，編集はCP／Mマイコンで稼動し，実行は大型計算機で行ない，結果をまたCP／Mのファイルにしまうことが可能になります。

TSS時のデータフォーマットは，データ7ビット，パリティ偶数，ストップビット1ビットとなっていますので，初期設定プログラムは次のようになります。

```
RSSTAT  EQU  61 H
RSDATA  EQU  60 H
        LD  A,01000000 B；リセット
        OUT  (RSSTAT),A
        LD  A,01111010 B；モード
              設定，クロック×16
        OUT  (RSSTAT), A
        LD  A,00110111 B；ブレークキャ
              ラクタ0
        OUT  (RSSTAT),A
```

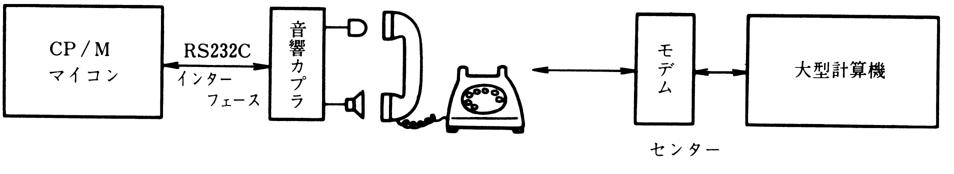
データ受信プログラムは次のようになります。

```
RSIN  IN  A,(RSSTAT)；
      AND  00000010 B；RxRDY？
      JP  Z,RSIN
      IN  A,(RSDATA)；データ入
              力
      RET
```

同様にして送信プログラムは次のようになります。

```
RSOUT    PUSH AF；データセーブ
RSOUT 1  IN A,(RSSTAT)；Tx RDY？
         RL  A
         JP  NC,RSOUT 1
         POP  AF
         OUT  (RSDATA),A；
              データ出力
```

なお，CP／Mマイコンをインテリジェントターミナルにするプログラムは参考文献[4-6]を参照して下さい。



付図G　大型計算機のＴＳＳ端末としての利用

第1部第7章

基本回路のテストとモニタの作り方

7

ここでは，前章までで作製した手作りコンピュータの基本回路が，完全に作動するかを調べるテスト方法について述べます。

また，次章以降でソフトを導入するための基本になるモニタプログラムをEPROMに書き込みます。

第7章　基本回路のテストとモニタの作り方

7.1 基本回路のテスト方法

本章では，第5章で作った各基本回路の配線ずみ基板に対する簡単な動作テストの方法について述べます。書込みが完全に行なえる手動のPROMライタを用いて，EPROM 2716に簡単なプログラムを書き込み，プログラムに従った動作が行なわれるかをチェックします。各種のテストプログラムを書き込みますので，あらかじめEPROM 2716を3個以上準備しておいて下さい。

7.1.1　CPU基板のテスト方法

はじめに，CPU基板とキーボード基板のテスト方法について述べます。使用するすべてのICを挿入する前に，CPU基板およびキーボード基板に正しく5V電源が供給されているかを確認してから，ICを挿入します。

CPU基板に対しては巻末折込みの図4.1のCPU図面を参照しながら，以下の点を確認して下さい。

① クロック4MHzが発振しているか

② 外部リセットスイッチ（図4.1の**1B**）を押すことにより，**図7.1 (a)** のような$\overline{\text{RESET}}$が出力されるか。ただし，このパルス幅は狭いので，シンクロスコープによっては観測されないことがあります。その場合は単安定フリップフロップ74121のB入力（5番ピン）が，リセットスイッチを押すことにより必ず“High”から“Low”になることを確認して下さい

③ $\overline{\text{BANK SELECT}}$信号が“High”状態となっているか。PROM 2716を用いてプログラムチェックを行ないますので，ROM領域でプログラムを動作させるために，PROM選択信号である$\overline{\text{BANK SELECT}}$信号が“High”状態となっていることを確認します

④ 割込み信号入力$\overline{\text{INT}}$（Z-80の16番ピン），$\overline{\text{NMI}}$（同17番ピン）および$\overline{\text{BUSRQ}}$（同

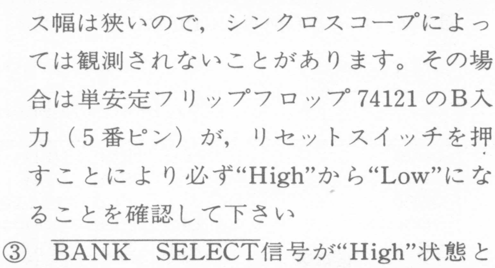


図7.1　$\overline{\text{RESET}}$信号の波形

25番ピン）が“High”状態になっているか確認して下さい。

同様にして，キーボード基板で，外部リセットスイッチを押すことにより，図7.1(b)に示すような$\overline{\text{RESET}}$信号が8255 Aのリセット入力35番ピンに入力されることを確認して下さい。

■**CPU動作の確認**

実際にプログラムを走らせることにより，マイクロプロセッサZ-80が動作するかどうかをテストします。はじめに，**図7.2**のようなもっとも簡単なプログラムであるNOP（No Operation）の繰返し動作によって，CPU動作の確認を行ないます。

PROMに機械語を書き込み，CPU基板に挿入し，電源を入れてリセットスイッチを押して下さい。これにより，正しくPROMの最初の命令から実行され，このとき以下の信号が出力されているかどうかを確認して下さい。

$\overline{\text{RAS}}$信号（コネクタA 20番端子）………
　不規則な繰返しクロックパルス

$\overline{\text{CAS}}$信号（コネクタB 20番端子）………
　不規則な繰返しクロックパルス

$\overline{\text{MEMW}}$信号（コネクタA 18番端子）……
　つねに“High”状態

$\overline{\text{MEMR}}$信号（コネクタB 18番端子）……
　不規則な繰返しクロックパルス

$\overline{\text{IOW}}$信号（コネクタA 19番端子）………
　つねに“High”な状態

$\overline{\text{IOR}}$信号（コネクタB 19番端子）………
　つねに“High”状態

以上の信号が出力されていなければ，CPU基板内の配線に誤りがあると思われます。基板の配線を1本1本確認して下さい。

■**機械語リストの読み方**

EPROMに書き込むべきプログラムを，本書では16進数字だけを羅列したダンプリストで

```
                           ; CPU TEST
0000 00           TOP:     NOP
0001 F20000                JP TOP
0004                       END
```

図7.2　もっとも簡単なCPU動作プログラム

```
LOC.   OBJ                 SOURCE STATEMENT

                ;TEST DRAM PROGRAM
                ;
007F            COMMD    EQU   7FH          ;
007D            PB       EQU   7DH          ;
00F0            OK       EQU   11110000B    ;
000F            ERROR    EQU   00001111B    ;
0800            TOPADR   EQU   800H         ;
アドレス  機械語  ;        アセンブラ言語
0000   210008   DRAM     LD HL,TOPADR       ;
0003   3E91              LD A,91H
0005   D37F              OUT (COMMD),A
0007   44       D64K     LD B,H             ;
0008   46                LD B,(HL)          ;
0009   78                LD A,B             ;
000A   BE                CP (HL)            ;
000B   C21F00            JP NZ,OERROR       ;
000E   23                INC HL
                ;
000F   7C                LD A,H             ;
0010   FE00              CP 0               ;
0012   C20700            JP NZ,D64K         ;
0015   7D                LD A,L             ;
0016   FE00              CP 0               ;
0018   C20700            JP NZ,D64K         ;
001B   3EF0              LD A,OK            ;
001D   D37D              OUT (PB),A
001F   3E0F     OERROR   LD A,ERROR         ;
0021   D37D              OUT (PB),A
0023   C32300   LOOP     JP LOOP
                ;
0026                     END
```

コメント文
EQU文
（コメント文・EQU文）プログラムを見やすくするための文
END文 ― プログラムの終わりを示す

図7.3　ザイログのZ-80アセンブラリストの見方

```
LOC.  OBJ               SOURCE STATEMENT

0000  3E91      TOP:    LD A,91H      ;モードセレクト
0002  D37F              OUT (7FH),A   ;PORT A & C0..C3 IN
                                      ;PORT B & C4..C7 OUT
0004  3E00              LD A,0
0006  D37D              OUT (7DH),A   ;PORT B 0 OUT
0008  3EFF              LD A,0FFH
000A  D37D              OUT (7DH),A   ;PORT B FFH OUT
000C  C30000            JP TOP
```

図 7.4 ポートBに矩形波を出力するプログラム

はなく，図7.2のようにアセンブラ形式で示します。このリストは，番地，機械語およびアセンブラリストを対応させて同時に表示したものです。

このリストは，1命令ごとに配置番地と機械語および命令内容が明瞭に示されていますので，EPROMへの書込み作業に適した構成となっています。このアセンブラリストを理解することで，一部の命令の変更が機械語レベルでも可能になります。

■コメントとEQU文

このアセンブラリストでは，EPROMに書き込む機械語のほか，コメントやEQUと呼ばれる数値をラベル名で指定する文が挿入されています。例えば**図 7.3** のアセンブラリスト中の□で囲んだ部分はコメント文で，⬚で囲んだ部分はEQU文です。

EPROMに書き込むときは，これらの文は無視し，LOCで示されたアドレスに対しOBJ欄に示す機械語を書き込んで下さい。その他，プログラムの配置位置を示すLOC文や，アセンブラの終わりを示すEND文も無視して下さい。

■ 8255 Aの動作テスト

CPU動作の確認が終了したら，次にキーボード制御用のインターフェースIC，8255 Aの動作の確認を行ないます。

図 7.4 に示すプログラムを，新しくPROMに書き込み，CPU基板に挿入します。このテストプログラムは，8255 AのBポートの8ビットすべてに対して，**図 7.5** に示す矩形波を出力します。

プログラムは，BポートとCポートの上位4ビットを出力モードに設定し，次にBポートに対して，データ“00”と“FF”を繰り返し出力するループとなっています。したがって，Bポートの端子をオシロスコープなどで観測して，矩形波が出力されていれば，8255 Aの動作テストは終了します。もし矩形波が発生されていなければ，以下の点を確認して下さい。

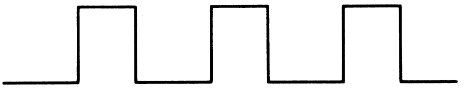


図 7.5 ポートBすべての端子に矩形波形が出力される

Bポート出力が“Low”レベルで変化しないときは，インターフェース基板と，各ICに5V電源が供給されているか否かを確認して下さい。

また，Bポート出力がつねに“Low”レベルの場合は，CPUのクロックの4MHzに，EPROM 2716 (450 ns) が追従しないことが原因とも考えられます。この場合は，CPU Z-80のM1サイクルごとに正しく$\overline{\mathrm{WAIT}}$信号が発生して，その信号が$\overline{\mathrm{WAIT}}$端子（24番ピン）に印加されているかどうかを確認して下さい。

次に，Bポート出力が“High”で一定の場合は，8255 Aがリセット状態で，すべてのポートが入力状態にあると考えられます。このときは正しくプログラムが動作していないことが原因と考えられますので，プログラムがEPROMに正確に書き込まれているかどうかを再確認する必要があります。

プログラムが正しく書き込まれているなら，次にCPU基板から発生する$\overline{\mathrm{IOW}}$，$\overline{\mathrm{IOR}}$，A_0，A_1が8255 Aに供給されているはずです。したがって，これが確実に加えられていることを確認する必要があります。このとき，はじめにCPU基板上での信号の波形を観測し，これとまったく

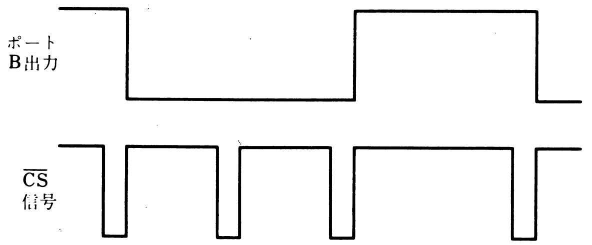


図 7.6　正常な$\overline{\mathrm{CS}}$信号の様子

同形の波形が8255 A上の入力部分に供給されていれば，正しい状態であると考えられます。

また，8255 Aのチップセレクト端子$\overline{\mathrm{CS}}$に，図 7.6 に示すポートB出力に同期した矩形波が入力されていることを確認します。

以上の信号が正しく与えられていて，かつ信号が正常に出ていないときには，原因は配線のミスと考えられます。基板上や本体コネクタ端子部分の配線を再確認しなければなりません。

7.1.2　キーボード基板のテスト方法

8255 Aの出力動作が正しく実行されていることが確認できたら，8255 Aの入力動作のテストを兼ねて，キーボードの動作テストを行ないます。

■キーボードの動作プログラム

図 7.7 のキーボードテストプログラムを実行させます。このプログラムは，キーボードから1文字読んでその内容をBポートに出力するプログラムです。キーボードを押した後，オシロスコープでBポートのPB_0～PB_7を観測し，キーインした文字に対応するアスキーコードが出力されていれば，キーボードが正しく動作していることになります。

ここでは図 7.8 に示すように，Cポートの0ビットに入力したキーボードのストローブ信号が“High”から“Low”に変化した時点をとらえて，Aポートからキーボードのデータを読み込

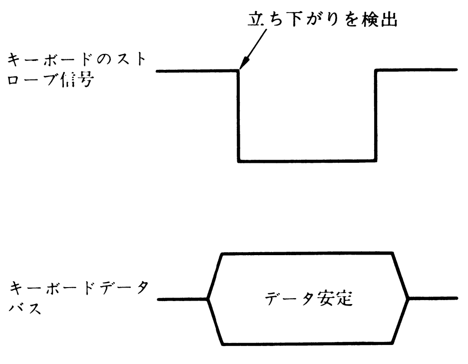


図 7.8　キーボードのデータ読込み

```
LOC.  OBJ                SOURCE STATEMENT

                  ;
                  ; CONSOLE INPUT PROGRAM
                  ;  KEYBOARD ---> (A REG.)
0000  3E91        CONIN:  LD A,91H      ;MODE SELECT OUT
0002  D37F                OUT (7FH),A
0004  DB7E        STBO:   IN A,(7EH)    ;INPUT FROM PORT CO..C3
0006  CB47                BIT 0,A        ;LOOP UNTILL STROBE=1
0008  CA0400              JP Z,STBO     ;
000B  DB7E        STB1:   IN A,(7EH)    ;INPUT FROM PORT CO..C3
000D  CB47                BIT 0,A       ;LOOP UNTILL STROBE=0
000F  C20B00              JP NZ,STB1
                  ;
0012  DB7C                IN A,(7CH)    ;INPUT FROM A PORT
0014  00                  NOP           ;
0015  E67F                AND 7FH       ;MASK MSB
0017  D37D        OUTB:   OUT (7DH),A
0019  C30400              JP STBO
```

図 7.7　キーボード入力をそのままポートBに出力するプログラム

み，Bポートに出力するように，テストプログラムを構成しています。

購入したキーボードの仕様により，もしアスキーコードが“High”，“Low”反転で，データ入力される場合は，14 H番地にある命令NOPの00を，CPLの2Fと変えて下さい。

■エラー対策

キーボードで入力したコードが出力されない場合は，キーを押したときに正しくストローブ信号がキーボードから出力されているか，また

《EPROM書込み誤りの対策》

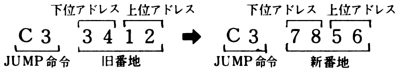
EPROMへ手動で書込み作業中，誤って書き込んでしまうと，ふたたび全部を消去してから，書き直さなければなりません。この場合の対策について述べます。

■NOPの利用

EPROMは，消去状態ではFFHになっています。書き込むことにより，“High” から “Low” へ対応するビットが変化しますが，いったん，“Low” が書込まれてしまったビットに “High”を書き込んでも，ふたたび “High” にもどることはありません。

そこで，**付図7.1**に示すように誤って書込みを行なってしまった場合，0番地からその番地までのすべてに “00” (No Operation) を書込み，その次の番地から命令の書込みを行なえばいいことになります。しかし，このとき注意しなければならないことは，JUMPやサブルーチン命令の飛び先番地を変更しなければならないことです。変更は次のように行ないます。

例えば，1234番地への無条件JUMPを5678番地へのJUMPに変更するときは，次のようになります。



同様に，

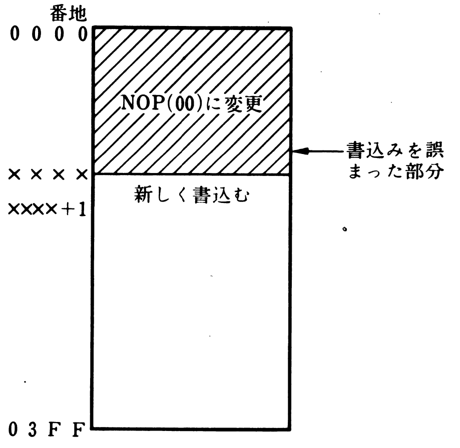
CAL (CD)，JNZ (C2)，JZZ (CA)，

など主命令も飛び先番地を変更しなければなりません。

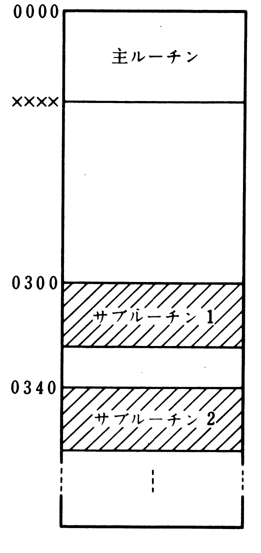
とくにアドレス設定の際に，下位アドレスが先で，次に上位アドレスが配置されることをまちがえないようにして下さい。

■サブルーチンの多用

書込み誤りによる被害を最小限におさえるには，サブルーチンの利用が有効です。多くのサブルーチンによりプログラムを構成し，各サブルーチンは**付図7.2**に示すように，EPROMの後半部分に配置しておきます。そのようにしてメインルーチンを先頭番地から書き込めば，もし誤って書き込んだ場合には，前述のようにその部分だけをNOPに書き換えて，ふたたび次の番地から書き込めばいいことになり，訂正部分が少なくてすみます。



付図7.1　先頭番地から誤った番地までをNOPに変更する場合



付図7.2　サブルーチンの使用

アスキーコードが正しく出力されているかを確認して下さい。

また，インターフェースIC　8255 AへのCPUからの制御信号は，キーボードストローブ入力待ち状態で，**図7.9**に示すような波形で$\overline{IOR}$，$\overline{CS}$信号が発生しているかどうかを確認して下さい。この場合，$\overline{IOW}$信号は，つねに“High”の状態を保っていなければなりません。

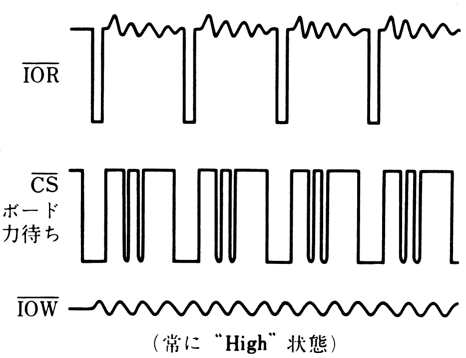


図7.9　正しいCPU制御信号の波形

以上の信号が正しく発生していて，すべてのBポートが“Low”に固定されているときは，CPUの24番ピンに$\overline{WAIT}$信号が入っているかどうか，またはCPU基板の$\overline{BANK\ SELECT}$信号が“High”でEPROMが選択されているかどうかをふたたび確認して下さい。

7.1.3　CRT基板のテストプログラム

CRT制御基板の動作を，CRTディスプレイ画面の左上部分に英文のA，Bの2文字を表示させてテストします。

テストプログラムを実行させる前に，CRT制御基板上の水晶発振回路が正しく8 MHzで発振し，シフトレジスタLS 166のCP（7番ピン）に供給されていて，かつ1 MHzがHD 46505の21番ピンに供給されていることを確認しておいて下さい。

■テストプログラムの実行

図7.10に示したテストプログラムのリストをEPROMに書き込んで下さい。このプログラムは，前半部分でCRT制御用の専用IC・HD 46505のパラメータの設定を行ないます。

プログラムの最後の部分で，AおよびBのキーコードをVRAMの開始番地F 000 HおよびF 001 Hにストアし，これにより画面左上にAとBが表示されるようにしています。設定する各パラメータの意味は，第4章を参照して下さい。

書込みが終了したEPROMをCPU基板に挿入し，電源を入れてリセットスイッチを押すと，正しく配線されていれば**図7.11**に示すように縦20文字×横40文字の合計800文字が表示され，かつ左上に文字A，Bが表示されるはずです。このときラスタ信号さえも出ないならば，HD 46565に対する各レジスタのパラメータ書込み動作が正しく実行されていないので，ふたたびCRT制御基板の配線チェックをして下さい。

画面に何も表示されていないときは，図4.13の配線図面中**7D**，**8D**にある映像信号の合成回路の不良ということが考えられます。このときは，水平同期信号（Hsync），および垂直同期信号（Vsync），映像信号（LS 166のQ_H出力）が正しく出力されているかどうかを確認して下さい。

■表示文字の誤り

20字×40字の合計800文字は表示されますが，左上の表示が指定した文字であるA，Bでないときは，VRAMのスタートアドレスの設定誤りの可能性があります。

この場合は，同じく図4.13の配線図面中の**1A**，**2A**にあるVRAMアドレス設定部分が正しく配線されていないためとも考えられます。再度チェックして下さい。

■HD 46505のパラメータ設定

水平同期信号Hsyncおよび垂直同期信号Vsyncが出力されていない場合は，HD 46505に対するパラメータの設定が正しく行なわれていないことが原因です。ソフトのエラーとして，プログラムが正しくEPROMに書き込まれているかどうかを確認して下さい。ハードのエラーとしては，HD 46505に対する$\overline{CS}$信号，$\overline{IOW}$信号が発生しているかどうかを確めて下さい。

7.1.4 64 K DRAMのテスト方法

64 Kのダイナミック RAM 配線基板の READ／WRITEテストを行なうには，DRAM 領域の0800 H番地からFFFFH番地までのアドレスに，データを書き込んだ後，そのデータを読み出してみて，正しくREAD／WRITEが

```
LOC.   OBJ                 SOURCE STATEMENT
                   ;CRTC HD46505 TEST PROGRAM
                   ;
F000               VRAM    EQU    0F000H      ;TOP OF THE VRAM ADDRESS
0000   3E00        CRTC    LD A,0
0002   47                  LD B,A             ;B<-A
0003   211F00              LD HL,CRPR0        ;TOP OF THE TABLE
0006   D3F6        CRTC1   OUT (0F6H),A       ;レジスタセット
0008   7E                  LD A,(HL)          ;
0009   D3F7                OUT (0F7H),A       ;レジスタライト
000B   23                  INC HL
000C   04                  INC B
000D   78                  LD A,B
000E   FE0E                CP 14              14回繰り返す
0010   38F4                JR C,CRTC1
0012   3E41                LD A,'A'           }
0014   3200F0              LD (VRAM),A        } DISPLAY 'AB'
0017   3E42                LD A,'B'           }
0019   3201F0              LD (VRAM+1),A      }
001C   C31C00      LOOP    JP LOOP            ;END OF THE PROGRAM
001F   3F          CRPR0   DEFB 63            ;水平総文字数
0020   28          CRPR1   DEFB 40            ;水平表示文字数
0021   32          CRPR2   DEFB 50            ;水平同期位置
0022   04          CRPR3   DEFB 4             ;水平同期パルス幅
0023   19          CRPR4   DEFB 25            ;垂直総文字数
0024   01          CRPR5   DEFB 1             ;トータルアジャスト
0025   14          CRPR6   DEFB 20            ;垂直表示文字数
0026   16          CRPR7   DEFB 22            ;垂直同期位置
0027   00          CRPR8   DEFB 0             ;ノンインターレースモード
0028   09          CRPR9   DEFB 9             ;最大ラスタアドレス
0029   65          CRPR10  DEFB 65H           ;カーソルスタートラスタ
002A   08          CRPR11  DEFB 8             ;カーソルエンドラスタ
002B   00          CRPR12  DEFB 0             ;VRAMスタートアドレス(上位)
002C   00          CRPR13  DEFB 0             ;VRAMスタートアドレス(下位)
```

図 7.10 CRT制御基板のテストプログラム

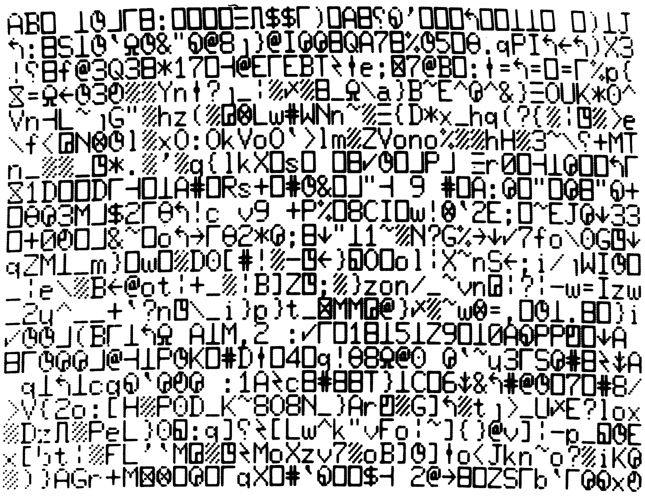


図 7.11 A，Bの表示例

```
LOC.  OBJ                SOURCE STATEMENT

                    ;TEST DRAM PROGRAM
                    ;
007F                COMMD   EQU  7FH        ;
007D                PB      EQU  7DH        ;
00F0                OK      EQU  11110000B  ;
000F                ERROR   EQU  00001111B  ;
0800                TOPADR  EQU  800H       ;
                    ;
0000  210008        DRAM    LD HL,TOPADR    ; メモリテスト開始番地設定
0003  3E91                  LD A,91H        } 8255Aのモード設定
0005  D37F                  OUT (COMMD),A   }
0007  44            D64K    LD B,H          ;} Hレジスタの内容をメモリに書込み
0008  70                    LD (HL),B       ;}
0009  78                    LD A,B          ; メモリ内容を読込み
000A  BE                    CP (HL)         ; 照合
000B  C22200                JP NZ,OERROR    ; 合致しなければエラー表示ルーチンへ
000E  23                    INC HL
000F  7C                    LD A,H          ;}
0010  FE00                  CP 0            ;}
0012  C20700                JP NZ,D64K      ;} アドレス800HからFFFFHまでメモリテストを繰り返す
0015  7D                    LD A,L          ;}
0016  FE00                  CP 0            ;}
0018  C20700                JP NZ,D64K      ;}
001B  3EF0                  LD A,OK         ;} すべて正常の場合はF0Hを出力
001D  D37D                  OUT (PB),A      }
001F  C32600                JP LOOP
0022  3E0F          OERROR  LD A,ERROR      ;} エラーが生じた場合は0FHを出力
0024  D37D                  OUT (PB),A      }
0026  C32600        LOOP    JP LOOP
```

図 7.12　DRAMテストプログラム

行なわれているかどうかをテストします。

■テストプログラム

64K DRAMのテストプログラムを**図 7.12**に示します。このプログラムでは，CPUのHレジスタの内容をRAM領域にREAD／WRITEし，これが正しく動作を終了すれば，8255AのBポートにデータF0Hが出力されます。しかし，1個所でも正しくREAD／WRITEされないアドレスが存在すれば，Bポートに0FHが出力されます。

プログラムが正しく動作し，正常に終了した場合は，8個のDRAMの任意の1個を引き抜いてプログラムを動作させ，エラーを発生させ，Bポートに0FHが出力されることを確かめて下さい。

■エラー対策

DRAMが正常にREAD／WRITEされない場合は，プリント基板の配線チェックを行なって下さい。このDRAM基板は，単純な構成なので，正しく配線が施されていれば，確実に動作するはずです。入念に配線チェックを行なってみましょう。

配線が正しく，かつRAS，CAS信号が入力されていても，正しくREAD／WRITEされない場合は，DRAMの破損によることも考えられますので，新しいDRAMに入れ換えて同じプログラムの動作のテストを行なって下さい。

7.2 自動EPROMライタ回路のテスト方法

第7章　基本回路のテストとモニタの作り方

7.1節の各基板が正しく動作したら，計算機としての機能は一応整ったことになります。

この手作りコンピュータを利用した応用の第一歩として，自動によるEPROMライタを動作させせることにします。

この自動EPROMライタが動くと，手書きに

比べて飛躍的にプログラム書込み能率が向上します。

7.2.1 自動EPROMライタ回路の構成

EPROMライタの回路図は，図6.5に示しました。このライタは，自動と手動の両方を兼用した回路で，自動の場合は同図面中**4A**に配置したインターフェースIC 8255 Aを介して，アドレス設定，データ設定および書込みタイミングパルス発生を行ないます。

8255 Aの3つのポートは，次のように割り当てています。

〈データ設定〉

データ	Bポート
D_0〜D_7	PB_0〜PB_7

〈アドレス設定〉

下位アドレス	Aポート
A_0〜A_7	PA_0〜PA_7
上位アドレス	Cポート
A_8〜A_{11}	PC_0〜PC_3

〈書込みパルス発生〉

Cポート	PC_5，PC_6

〈READ／WRITE制御〉

Cポート	PC_4

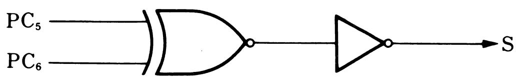
8255 Aの全ポートを出力モードにしてデータおよびアドレス設定の後，PC_5を“High”から“Low”へ状態を変化させることにより，手動と同じタイミングで1バイトのデータが書き込まれます。

ただし，この自動操作の場合は，PROMをソケットに挿入後，+25 V電源スイッチはON状態にしておかなければなりません。

■書込みパルスの発生

書込みタイミングパルスは，CポートのPC_5およびPC_6を用いて発生しています。ここでPC_5とPC_6の2本のポートを利用している理由は，**8255 Aのモード切換え動作時における書込みパルス発生を阻止する**ためです。

モード設定動作時はすべてのポートが“Low”に設定されてしまいます。このためEPROMへの書込みと，EPROMの内容の読込み作業を繰り返す場合，Bポートの入出力のモード切換え動作を行なわなければならず，このとき書込みパルスが誤って発生する恐れがあるからです。



	PC_5	PC_6	S
RESET状態	1	1	1
モード切換え動作時	0	0	1
書込み動作時	0	1	0

書込み動作時のみパルスを発生

図7.13 PC_5，PC_6の2出力を用いた書込みパルスの発生方法

このように，モード切換え動作時も単安定回路LS 123（図6.5の**2C**）の入力信号を“High”に保つために，**図7.13**に示すようにPC_5とPC_6の信号に対するEXORとNOT回路を挿入しています。この回路が図6.5上**3C**の部分です。

■WRITE／READ切換え

この回路は，EPROMライタの書込み動作だけでなく，EPROMの書込み内容の読込み作業も行なえるように構成してあります。このREADとWRITEの切換えがCポートのPC_4（PW／$\overline{PR}$信号）で，図6.5上**4C**に配置した双方向トライステートバスバッファ74 LS 245の方向DIRを制御しています。

このとき，PC_4が“High”でA→Bすなわち，8255 AからEPROM 2716へデータが送られて書込み動作が行なわれます。また逆に，PC_4が“Low”のときはB→Aとなり，2716のデータ出力が8255 AのBポートを介して読み込まれることになります。

■ 8255 Aの操作

8255 Aはキーボードの制御用にも利用しています。この2つの8255 Aが同時に動作しないように制御するのが$\overline{CS}$（Chip Select信号）です。EPROMライタ用の$\overline{CS}$信号を発生しているのが，図6.5上**2A**に配置してあるデコーダ74 LS 42であり，デコーダ出力は1番ピンの$\overline{0}$を利用してるので，デコーダ入力であるアドレス

```
LOC.  OBJ              SOURCE STATEMENT
-----------------------------------------------------------------------
                       ;
                       ;PORT OF THE PROM WRITER PROGRAM
                       ;
000C                   PA       EQU  0CH          ;
000D                   PB       EQU  0DH          ;
000E                   PC       EQU  0EH          ;
000F                   COMMD    EQU  0FH          ;
0080                   SETPPI   EQU  10000000B    ;
                       ;
-----------------------------------------------------------------------
0000  3E80             TPPI:    LD A,SETPPI       ;PROM8255Aモード選択
0002  D30F                      OUT (COMMD),A     ;
0004  3E00             TOP      LD A,00000000B    ;データ"0"セット
0006  D30C                      OUT (PA),A        ;Aポート出力
0008  D30D                      OUT (PB),A        ;Bポート出力
000A  D30E                      OUT (PC),A        ;Cポート出力
000C  3EFF                      LD A,11111111B    ;データ"FF"セット
000E  D30C                      OUT (PA),A         Aポート出力
0010  D30D                      OUT (PB),A         Bポート出力
0012  D30E                      OUT (PC),A         Cポート出力
0014  18EE                      JR TOP
```

図 7.14　PROM用の8255 A動作テストプログラム

A_7～A_4をすべて0と指定した場合，この8255 Aが選択されることになります。

7.2.2　自動EPROMライタのテスト方法

自動EPROMライタ回路が正しく動作するかを調べるテスト方法について述べます。このテストによって8255 Aがプログラム通りに動作すれば，手動ではROMライタとして完動していますので，次節のプログラムで自動EPROMライタが完成することになります。

■ 8255 Aの動作テスト

はじめに，EPROM操作用の8255 Aが正しく選択され，各ポートの動作が正常に行なわれるかどうかの動作テストを行ないます。テストプログラムのリストを**図 7.14** に示します。このプログラムを手動で2716に書き込んで下さい。

EPROMライタ基板上に74 LS 245，74 LS 42，8255 Aだけを挿入し，アドレス設定やデータ設定用のすべてのDIPスイッチをOFF状態にしてプログラムを動作させて下さい。このとき，すべてのポートから**図 7.15** に示すような矩形波が出力されれば，8255 Aが正しく動作していることになります。また，正常に動作していれば，アドレス表示用のLEDにも矩形波が入力されますので，発光状態となります。

■エラー対策

各ポートに矩形波が出力されない場合は，ふたたび配線チェックを行なって下さい。また，手動状態でのEPROM読出し動作で，正しくプログラムが書き込まれているかどうかのチェックも行なって下さい。

以上のチェックが終了したら，8255 Aの制御入力である$\overline{CS}$に対し，図 7.15に示したような$\overline{CS}$信号波形が入力されているかを確認します。そのとき，他の制御入力のA_0，A_1，$\overline{IOW}$，$\overline{IOR}$の各信号が，CPU基板上から出力されているときの波形と同形になっているかどうかも確めて下さい。

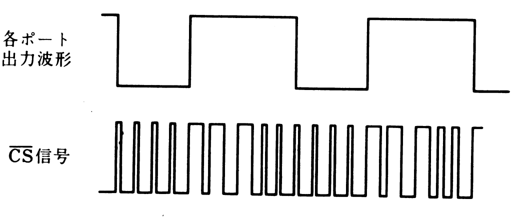


図 7.15　各ポートの出力波形と$\overline{CS}$信号

第７章 基本回路のテストとモニタの作り方

7.3 基本モニタプログラムの作り方

EPROMの自動書込みが行なえることが確認できたら，次にモニタプログラムの作製を行ないます。モニタプログラムは，対話形式でキーボードからメモリにプログラムやデータを書き込んだり，読出しを行なったり，またメモリの内容をEPROMに自動書込みを行なうものです。

7.3.1 EPROMの自動書込み

7.1節ではすべてのプログラムを，DIPスイッチを用いて手動でEPROMに書き込んでいました。モニタプログラムは，500 Hバイト程度の長いプログラムなので，手動で書込みは不可能です。そこで，キーボードとの対話形式で機械語をRAMに一時記憶させ，その内容を自動的にEPROMに書き込むEPROM自動書込みプログラムを作製します。

ここでは，最終的なモニタプログラムを製作するまでの最初の段階として，基本モニタプログラムを作成します。

■基本モニタプログラム

第１段階は，もっとも原始的な自動EPROM書込みプログラムとして，キーボードから機械語を入力し，その内容をEPROMに書き込む作業を行なうモニタを作ります。

原始プログラム自体は，もちろん，手動でEPROMに書き込まなければなりません。このプログラムリストを**図7.16**(次ページ)に示しますが，52 Hバイトの短いプログラム構成となっています。また，基本モニタプログラムのフローチャートは**図7.17**のようになっています。

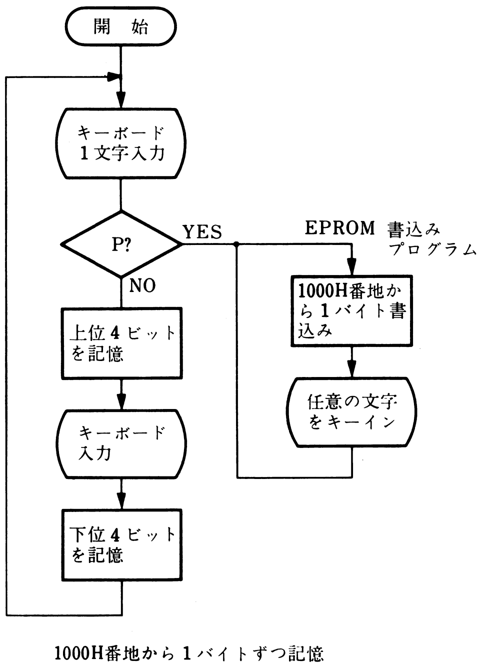
すなわち，この基本モニタの動作内容は，EPROMに書き込むべきプログラムを1000 H番地から１バイト（16進数２桁）ずつキーボードを用いて入力し*，プログラムのキーボード入力が終わったら，“P”を押せば，EPROMに２Kバイト分（1000 H～17 FFH番地）のプログラムが順次に書き込まれるように構成しています。

■基本モニタプログラムの使い方

図7.16に示した基本モニタプログラムをEPROMに手動で書込み，それができたら，このEPROMをCPU基板のEPROMソケットに挿入します。次に，EPROMライタ基板のDIPスイッチをすべてOFF（自動モードの設定）にして，以下に述べる順序で自動書込みを行なって下さい。

① EPROM基板の５Ｖ電源SW1をONにする
② RESETスイッチを押す
③ 機械語リストを見ながら機械語を，0

* 7.1.2項で述べたように，正論理でアスキーコードが入力される場合は，2 DH番地のCPLをNOP（00）に変更して下さい（81ページ参照）。



1000H番地から１バイトずつ記憶
0～9およびA～Fのアスキーコード２桁入力

図7.17 第１段階のEPROM自動書込みプログラムのフローチャート

～9とA～Fの16進2桁でキーボードから入力する。同時に内容がLEDに表示される

④ すべてのプログラムのキーインが終了したら，EPROMライタ基板上の5V電源のみを切り，消去ずみのEPROMを基板に挿入後5V電源，25V電源をONにする

⑤ “P”をキーインすることにより，最初の1バイトが書き込まれ，その内容がLEDに表示される

```
                    ;BASIC MONITOR PROGRAM
0000  3E91          TOP:    LD A,91H          ;} キーボード用8255Aモード設定
0002  D37F                  OUT (7FH),A       ;}
0004  3100A0                LD SP,0A000H      ;
0007  3E80                  LD A,80H          ;} PROMライタ用8255Aモード設定
0009  D30F                  OUT (0FH),A       ;}
000B  3E70                  LD A,01110000B    ;} C(4),……C(6)を用いたPROMライタモード設定
000D  D30E                  OUT (0EH),A       ;}
000F  210010                LD HL,1000H       ; メモリ開始番地設定
0012  CD2400        MON1:   CALL DKYIN        ; 1文字入力
0015  ED6F                  RLD               ; メモリにストア
0017  CD2400                CALL DKYIN        ; 1文字入力
001A  ED6F                  RLD               ; メモリにストア
001C  CD4E00                CALL DISP         } LED表示
001F  D30E                  OUT (0EH),A       }
0021  23                    INC HL            ;
0022  18EE                  JR MON1           ;
-------------------------------------------------------------------------
0024  DB7E          DKYIN:  IN A,(7EH)        ;} キーボードからの1文字入字
0026  CB47                  BIT 0,A           ;}
0028  28FA                  JR Z,DKYIN        ;}
002A  DB7E          KEY1:   IN A,(7EH)        ;}
002C  CB47                  BIT 0,A           ;
002E  20FA                  JR NZ,KEY1        ;
0030  DB7C                  IN A,(7CH)        ; Aポート入力
0032  00                    NOP
0033  E67F                  AND 01111111B     ; パリティビットを除く
0035  FE50                  CP 'P'            ; “P” ?
0037  2821                  JR Z,PROMW        ;
0039  FE58                  CP 'X'              “X” ?
003B  CA4400                JP Z,DECHL
003E  CB77                  BIT 6,A           ;} アスキーコードから16進数への変換
0040  C8                    RET Z             ;}
0041  C609                  ADD A,9           ;}
0043  C9                    RET               ;}
                    (0024～0043: サブルーチンキーボード入力 アスキー16進変換)
-------------------------------------------------------------------------
0044  2B            DECHL:  DEC HL            } “X”入力の処理
0045  CD4E00                CALL DISP         } 1番地もどす
0048  D30E                  OUT (0EH),A       }
004A  D1                    POP DE            }
004B  C31200                JP MON1
004E  7E            DISP:   LD A,(HL)         ;
004F  D30D                  OUT (0DH),A       ;  LEDディスプレイ用サブルーチン
0051  7D                    LD A,L            ;
0052  D30C                  OUT (0CH),A          アドレスおよびデータを表示
0054  7C                    LD A,H            ;
0055  E60F                  AND 00001111B     ;
0057  C6D0                  ADD A,11010000B   ;
0059  C9                    RET
-------------------------------------------------------------------------
005A  210010        PROMW:  LD HL,1000H         1000H番地を設定
005D  3E80                  LD A,80H          } PROMライタ用8255Aモード設定
005F  D30F                  OUT (0FH),A       }
0061  CD4E00        PROM1:  CALL DISP           アドレスおよびデータ出力
0064  E67F                  AND 01111111B       ライトパルス“Low”設定
0066  D30E                  OUT (0EH),A         Cポート出力
0068  CD2400                CALL DKYIN        ; キー入力待ち
006B  3E70                  LD A,01110000B    ; ライトパルス“High”設定
006D  D30E                  OUT (0EH),A       ;
006F  23                    INC HL            ;
0070  18EF                  JR PROM1          ;
                    (005A～0070: EPROMへの自動書込み)
-------------------------------------------------------------------------
```

図7.16　基本モニタプログラムリスト

⑥ スペースを1回キーインするごとに，1バイトずつ書き込まれる

⑦ 書込みが終了したら，まず25 VのSW2をOFFにし，次に5 VのSW1をOFFにして，EPROMをソケットからはずす

以上の操作により，キーボードからデータを入力して，EPROMにプログラムが書き込まれることになります。もし，③の機械語リストをキーインしているときに，誤まってキーインした場合は，“X”をキーインすると1番地分もどり，再入力可能となります。

■基本モニタのテスト

基本モニタを用いて簡単なEPROMの書込みテストを行なってみましょう。はじめに任意のデータ（例えばAAのようなデータ）を多数キーインしてみて，EPROMに書き込まれているかどうかを確かめて下さい。もし，正しく書き込まれていないときには，以下の点をチェックします。

① 少なくとも手動では正しく書込み動作ができるか

② EPROM 2716の21番ピンに，正しく25 Vが供給されているか

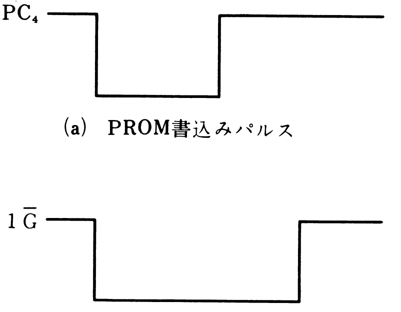
③ 74 LS 245の方向制御信号DIRが“High”となり，データがAからBの方向へ設定されているか

④ キーボードを押すごとに，CポートのPC_5端子から**図 7.18 (a)**に示す書込みパルスが発生しているか

⑤ 書込みパルスが発生したとき，74 LS 245のENABLE信号$\overline{G}$は，図 7.18 (b) に示す波形となっているか

以上の点が正しく動作するならば，書込みは正しく行なわれるはずです。それでもまだだめなときは，使用しているEPROMの以前のデータが完全に消去されていて，すべてのデータがFFHとなっているかどうかを確かめて下さい。

いままで述べた点がすべて正しければ，EPROM自体の書込み不良が原因です。EPROMは，インパルス状の電流であっても26 V以上の過電圧の印加で簡単に破壊されますので，取扱いには細心の注意を払って下さい。



(a) PROM書込みパルス

(b) バストランシーバ74LS245のENABLE信号

図 7.18 PC_4と$1\overline{G}$の波形

7.3.2 モニタプログラムの作り方

EPROMに書き込まれた基本モニタプログラムが正しく動作することが確認できたら，この基本モニタプログラムを用いて，ディスプレイを有効に活用したよりプログラム作成効率の高いモニタプログラムを作ります。

モニタプログラムのリストを**図 7.19**に示します。基本モニタプログラムを用いてキーボードから図 7.19のOBJ欄に示す16進数31，00，F 0，21……の順に入力し，消去ずみのEPROMに書き込んで下さい。

■モニタプログラムの機能

このモニタプログラムの構成を，**図 7.20**のフローチャートに示します。ディスプレイとの対話形式により，任意のメモリアドレスに対してREAD／WRITEおよび実行が行なえます。また，EPROMへの自動書込みおよび読込みがコマンドで行なえるようになっています。

使用できるコマンドを**表 7.1**に示します。キーインされたコマンド名と飛び先番地の参照は，モニタリストの1 EH番地から3 AH番地で行なっていて，“M”，“P”，“G”，“W”，“R”の5種類です。

ただし，このモニタは，第8章で詳しく述べるCP／Mマイコン用のモニタが追加可能となるように構成されています。したがって，そのとき追加するフロッピーディスクからの読込みコマンド“R”と書込みコマンド“W”が入っていますが，本章のモニタでは使用しません。

```
               LOC.  OBJ            SOURCE STATEMENT
                                    ;Z80 Monitor Program      コメントおよびアセンブラ文
               F000                 SPTOP     EQU  0F000H     スタックポインタ0F000Hに設定
               5AF3                 WBOOT     EQU  5AF3H      ;
               0000                 BANKST    EQU  0
               F000                 VSTAT     EQU  SPTOP      VRAMスタートアドレス
               F2F8                 VMLD      EQU  0F2F8H     CRTディスプレイ画面の最左, 最下行のアドレス
               0020                 NULL      EQU  ' '
               000D                 CR        EQU  0DH
               000A                 LF        EQU  0AH
               007F                 RUB       EQU  7FH
               0013                 CTRLS     EQU  13H
               0003                 CTRLC     EQU  03H
               0028                 LINE      EQU  40         ; 1行40文字
               0800                 VRMN      EQU  800H       ; 全VRAM領域文字数
               0050                 DISPROM   EQU  50H        ; EPROM切離し信号
               00F8                 DISK      EQU  0F8H       ;
               00F8                 DCOM      EQU  DISK       ; ディスクコマンドポート
               00F8                 DSTAT     EQU  DISK       ; ディスクステータスポート
               00F9                 TRACK     EQU  DISK+1     ; ディスクトラックポート
               00FA                 SECTP     EQU  DISK+2     ; ディスクセクタポート
               00FB                 DDATA     EQU  DISK+3     ; ディスクデータポート
               00FC                 WAIT      EQU  DISK+4     ; ディスクウエイトポート
               000A                 RTCNT     EQU  10 ;       ディスクエラー10回再試行
               000C                 PADR      EQU  0CH        ; ポートA  ┐
               000D                 PDATA     EQU  0DH        ; ポートB  │
               000E                 PMODE     EQU  0EH        ; ポートC  ├ 8255A
               000F                 PCOMMD    EQU  0FH        ; モード設定┘
               0080                 PROMPPI   EQU  10000000B  ;
               007C                 KEY       EQU  7CH        ;
               007E                 KEYSTAT   EQU  7EH        ;
               007E                 DPSTAT    EQU  KEYSTAT    ;
               007E                 DREADY    EQU  KEYSTAT    ;
               007F                 KCOMMD    EQU  7FH        ;
               0091                 KEYPPI    EQU  10010001B  ;
               F400                 HCOUNT    EQU  0F400H     ; 水平文字カウンタ
               F401                 BUFF      EQU  0F401H
               F402                 WORK1     EQU  0F402H
 E             F403                 WORK2     EQU  0F403H
 P             F4C0                 TRK       EQU  0F4C0H
 R             F4C1                 SECT      EQU  0F4C1H
 O             F4C3                 DMAADD    EQU  0F4C3H
 M             F4D0                 HLSF      EQU  0F4D0H     ヘッドロードフラグ
 ア            F4FD                 ERCNT     EQU  0F4FDH
 ド            F4FE                 SERCNT    EQU  0F4FEH     シークリトライカウンタ
 レ
 ス                                           ORG  0F800H     ;
0000-F800  3100F0    MOVEPGM:LD SP,SPTOP       ;  ┐
0003-F803  210000            LD HL,0           ;  │ バンク切換用のメモリ転送
0006-F806  1100F8            LD DE,MOVEPGM     ;  ├ EPROMの全領域(2048バイト分)をF800番地以後
0009-F809  010008            LD BC,2048        ;  │ に移す
000C-F80C  EDB0              LDIR              ;  ┘
000E-F80E  C311F8            JP MONITADR          転送後F811番地にJUMP
0011-F811  3E91      MONITADR:LD A,10010001B   ; ┐8255(A)のモード設定            ┐
0013-F813  D37F              OUT (KCOMMD),A    ; ┘C(0)~C(3)入力, C(4)~C(7)出力  │
0015-F815  3E50              LD A,DISPROM      ; ┐バンク切換え                  ├ EPROM領域切離し
0017-F817  D37E              OUT (DREADY),A    ; ┘                              ┘
0019-F819  CD40F8            CALL INIT         ; プログラム初期設定
001C-F81C  3E3E      MONIT1:LD A,'>'  ;OUTPUT A PROMPT MARK
                     ;*******************************  ┐
                     ; IF MONITOR THEN REMOVE ';'      │ モニタープログラムではCALL CONOTとし,
001E-F81E  CD23FA            CALL CONOT        ;       ├ 第9章のCP/M完成後はJP BOOT(C3EAFC)
                     ; IF CPM THEN REMOVE ';'          │ に変更する
                     ;       JP BOOT                   ┘
0021-F821  CDD2F9            CALL CONIN2       ;
0024-F824  FE4D              CP 'M'
0026-F826  CA4CF8            JP Z,MONITP       ;
0029-F829  FE50              CP 'P'
002B-F82B  CA53F9            JP Z,PROM         ;
002E-F82E  FE47              CP 'G'
0030-F830  CAB1F9            JP Z,JOBGIN       ;
0033-F833  FE57              CP 'W'
0035-F835  CA0CFB            JP Z,FDWRIT       ;
0038-F838  FE52              CP 'R'
003A-F83A  CA4AFB            JP Z,FDREAD       ;
```

図 7.19　モニタプログラムのリスト(1)

```
003D-F83D  C31CF8             JP MONIT1
0040-F840  CDD9F9     INIT:CALL CRTC              サブルーチン INIT
0043-F843  CDFEF9             CALL CREA           ; 全画面をクリア
0046-F846  3E00               LD A,0              ;
0048-F848  3200F4             LD (HCOUNT),A       ; 水平文字カウンタのリセット
004B-F84B  C9                 RET
004C-F84C  21C3F8     MONITP:LD HL,MMONIS         ; モニタプログラム開始
004F-F84F  CD0DFA             CALL FMSG           ;
0052-F852  CDCBF8             CALL D16IN          ; 16進4桁データをHLレジスタに入力
0055-F855  CD17FA     MONIP1:CALL CRLF            ; 復改および改行
0058-F858  CDF6F8             CALL MEMDIS         ; メモリディスプレイ開始 HL-5
005B-F85B  CDBCF9     MONIP2:CALL CONIN           ;
005E-F85E  FE20               CP ' '              ;
0060-F860  CA9CF8             JP Z,MONIP3
0063-F863  FE54               CP 'T'   ;
0065-F865  CA1CF8             JP Z,MONIT1         ;
0068-F868  FE58               CP 'X'
006A-F86A  CAAFF8             JP Z,MONIP6         ;
006D-F86D  FE47               CP 'G'
006F-F86F  D2B3F8             JP NC,MONIP7        ; 入力文字≧G?
0072-F872  FE2F               CP 2FH
0074-F874  DAB3F8             JP C,MONIP7         ;
0077-F877  32CEF0             LD (VSTAT+LINE*5+6),A   ; 画面5行目の6文字目に表示
007A-F87A  CB77               BIT 6,A             ;
007C-F87C  CA81F8             JP Z,MONIP4         ; 数字
007F-F87F  C609               ADD A,9             ;
0081-F881  ED6F       MONIP4:RLD                  ;
0083-F883  CDBCF9     MONI41:CALL CONIN           ;
0086-F886  FE47               CP 'G'              ; 入力文字 ≧G?
0088-F888  D2BBF8             JP NC,MONIP8
008B-F88B  FE2F               CP 2FH                入力文字 <0?
008D-F88D  DABBF8             JP C,MONIP8
0090-F890  32CFF0             LD (VSTAT+LINE*5+7),A   ; 画面5行目の7文字目に表示
0093-F893  CB77               BIT 6,A             ;
0095-F895  CA9AF8             JP Z,MONIP5
0098-F898  C609               ADD A,9             ;
009A-F89A  ED6F       MONIP5:RLD
009C-F89C  23         MONIP3:INC HL               ;
009D-F89D  E5                 PUSH HL             ;
009E-F89E  010D00             LD BC,13            ;
00A1-F8A1  09                 ADD HL,BC           ;
00A2-F8A2  CD10F9             CALL MEMDI2         ;
00A5-F8A5  21C8F0             LD HL,VSTAT+LINE*5  ; 画面5行目の1文字目にカーソル表示
00A8-F8A8  CDE2FA             CALL CARSOL         ;
00AB-F8AB  E1                 POP HL
00AC-F8AC  C35BF8             JP MONIP2
00AF-F8AF  2B         MONIP6:DEC HL               ; HL=HL-1  バックコード処理
00B0-F8B0  C355F8             JP MONIP1           ;
00B3-F8B3  3E3F       MONIP7:LD A,'?' ;
00B5-F8B5  32CEF0             LD (VSTAT+40*5+6),A  ;
00B8-F8B8  C35BF8             JP MONIP2           ;
00BB-F8BB  3E3F       MONIP8:LD A,'?' ;
00BD-F8BD  32CFF0             LD (VSTAT+LINE*5+7),A  ;
00C0-F8C0  C383F8             JP MONI41           ;
00C3-F8C3  0A0D4144 MMONIS: DEFB LF,CR,'A','D','R',LF,CR,0
00C7-F8C7  520A0D00
00CB-F8CB  D5         D16IN:PUSH DE               ; サブルーチン D16IN
00CC-F8CC  2103F4             LD HL,WORK2         ; キーボードから4桁16進データをHLレジスタに入力
00CF-F8CF  CDE9F8             CALL D16INA         ;
00D2-F8D2  ED6F               RLD                 ;
00D4-F8D4  CDE9F8             CALL D16INA         ;
00D7-F8D7  ED6F               RLD                 ;
00D9-F8D9  56                 LD D,(HL)           ;
00DA-F8DA  23                 INC HL              ;
00DB-F8DB  CDE9F8             CALL D16INA         ;
00DE-F8DE  ED6F               RLD                 ;
00E0-F8E0  CDE9F8             CALL D16INA         ;
00E3-F8E3  ED6F               RLD                 ;
00E5-F8E5  6E                 LD L,(HL)           ; (L)<……〔M〕
00E6-F8E6  62                 LD H,D              ; (H)<……D
00E7-F8E7  D1                 POP DE              ;
00E8-F8E8  C9                 RET
00E9-F8E9  CDD2F9     D16INA:CALL CONIN2          ; サブルーチン D16INA
00EC-F8EC  CB77       D16INB:BIT 6,A              ;   キーボード入力のアスキーコードから16進数へ変換
00EE-F8EE  CAF3F8             JP Z,D16INI         ;
00F1-F8F1  C609               ADD A,9             ;
```

図 7.19 モニタプログラムのリスト(2)

```
00F3- F8F3  E60F    D16INI:AND 00001111B    ;
00F5- F8F5  C9             RET
00F6- F8F6  E5      MEMDIS:PUSH HL            サブルーチン MEMDIS
00F7- F8F7  F5             PUSH AF          ;  HLレジスタの番地－5のメモリアドレスから
00F8- F8F8  C5             PUSH BC          ;  19番地分のメモリの内容を表示
00F9- F8F9  010500         LD BC,5          ;
00FC- F8FC  ED42           SBC HL,BC        ;
00FE- F8FE  0613           LD B,19
0100- F900  CD10F9  MEMDI1:CALL MEMDI2      ;
0103- F903  23             INC HL           ;
0104- F904  10FA           DJNZ MEMDI1      ;
0106- F906  21C8F0         LD HL,VSTAT+LINE*5   ;
0109- F909  CDE2FA         CALL CARSOL      ;
010C- F90C  C1             POP BC           ;
010D- F90D  F1             POP AF           ;
010E- F90E  E1             POP HL           ;
010F- F90F  C9             RET
0110- F910  3E20    MEMDI2:LD A,' ' ;
0112- F912  CD23FA         CALL CONOT       ; サブルーチン MEMDI2
0115- F915  CD25F9         CALL D16DIS      ;  1行分のメモリ内容の表示
0118- F918  3E20           LD A,' '         ;
011A- F91A  CD23FA         CALL CONOT       ;
011D- F91D  7E             LD A,(HL)        ;
011E- F91E  CD2EF9         CALL DHEXA       ;
0121- F921  CD17FA         CALL CRLF        ;
0124- F924  C9             RET
0125- F925  7C      D16DIS:LD A,H           ; サブルーチン D16DIS
0126- F926  CD2EF9         CALL DHEXA       ;  16進データの内容のアスキーコードによる表示
0129- F929  7D             LD A,L           ;
012A- F92A  CD2EF9         CALL DHEXA       ;
012D- F92D  C9             RET
012E- F92E  E5      DHEXA:PUSH HL           ; サブルーチン DHEXA
012F- F92F  2102F4         LD HL,WORK1      ;  8ビットの16進数の内容をアスキーコードによる2桁表示
0132- F932  77             LD (HL),A        ;
0133- F933  ED6F           RLD              ;
0135- F935  CD3FF9         CALL ASCII       ;
0138- F938  ED6F           RLD              ;
013A- F93A  CD3FF9         CALL ASCII       ;
013D- F93D  E1             POP HL           ;
013E- F93E  C9             RET
013F- F93F  E60F    ASCII: AND 00001111B    ;
0141- F941  FE0A           CP 10            ;
0143- F943  3004           JR NC,ALPA1      ; もし(A)>9ならALPA1へ
0145- F945  C630           ADD A,30H        ;
0147- F947  1806           JR ASCI1         ;
0149- F949  D609    ALPA1: SUB 9            ;
014B- F94B  E607           AND 00000111B    ;
014D- F94D  C640           ADD A,40H        ;
014F- F94F  CD23FA  ASCI1:  CALL CONOT      ;
0152- F952  C9             RET
0153- F953  210010  PROM:  LD HL,1000H      ; EPROMへのREAD/WRITEプログラム
0156- F956  CDD2F9         CALL CONIN2      ;
0159- F959  FE57           CP 'W'           ;
015B- F95B  CA8CF9         JP Z,WIN         ;
015E- F95E  3E82    PROMR: LD A,10000010B   ; ポートB入力
0160- F960  D30F           OUT (PCOMMD),A   ; モードセット
0162- F962  3EE0           LD A,0E0H        ; リードパルスHigh
0164- F964  D30E           OUT (PMODE),A    ; 設定
0166- F966  3EC0           LD A,0C0H        ; リードパルスLow
0168- F968  D30E           OUT (PMODE),A    ; 設定
016A- F96A  3EE0           LD A,0E0H        ; リードパルスHigh
016C- F96C  D30E           OUT (PMODE),A    ; 設定
016E- F96E  16FF           LD D,255         ;} 無駄時間  2msec×255≒0.5sec
0170- F970  CDFFFA         CALL TIME        ;}
0173- F973  7D      PROM2: LD A,L           ;
0174- F974  D30C           OUT (PADR),A     ; A(0)~A(7)出力
0176- F976  7C             LD A,H           ; (A)<……(H)
0177- F977  E60F           AND 00001111B    ;} リードモード設定およびリードパルスLow設定
0179- F979  C6E0           ADD A,11100000B  ;} 8255Cポートを使用
017B- F97B  D30E           OUT (PMODE),A    ;}
017D- F97D  DB0D           IN A,(PDATA )    ; データ入力
017F- F97F  77             LD (HL),A        ;
0180- F980  CD10F9         CALL MEMDI2      ;
0183- F983  23             INC HL           ;
0184- F984  7C             LD A,H           ;
0185- F985  FE18           CP 18H           ;
```

図7.19 モニタプログラムのリスト(3)

```
0187-F987  20EA              JR NZ,PROM2
0189-F989  C31CF8            JP MONIT1
018C-F98C  3E80      WIN:  LD A,10000000B     ; 全出力モード設定
018E-F98E  D30F              OUT (PCOMMD),A   ;
0190-F990  3E70              LD A,70H         ; C(4)~C(4)を使ってREAD/WRITEパルスセット
0192-F992  D30E              OUT (PMODE),A    ;
0194-F994  7E        PROM1: LD A,(HL)         ;
0195-F995  D30D              OUT (PDATA),A    ; データ出力　Bポート
0197-F997  7D                LD A,L           ;
0198-F998  D30C              OUT (PADR),A     ; A(0)~A(7)アドレス出力
019A-F99A  7C                LD A,H           ;
019B-F99B  E60F              AND 0FH          ;}
019D-F99D  C650              ADD A,50H        ;} Cポートを用いてライトモードおよびパルスLowセット
019F-F99F  D30E              OUT (PMODE),A    ;}
01A1-F9A1  CD10F9            CALL MEMDI2
01A4-F9A4  3E70              LD A,70H         ; パルス　High
01A6-F9A6  D30E              OUT (PMODE),A    ; 出力
01A8-F9A8  23                INC HL
01A9-F9A9  7C                LD A,H           ;
01AA-F9AA  FE18              CP 18H           ;
01AC-F9AC  20E6              JR NZ,PROM1      ;
01AE-F9AE  C31CF8            JP MONIT1        ;
01B1-F9B1  21C3F8    JOBGIN: LD HL,MMONIS     ; JOB実行プログラム
01B4-F9B4  CD0DFA            CALL PMSG        ;
01B7-F9B7  CDCBF8            CALL D16IN       ;
01BA-F9BA  E5                PUSH HL          ;
01BB-F9BB  C9                RET
01BC-F9BC  C5        CONIN: PUSH BC           ; サブルーチン　CONIN
01BD-F9BD  DB7E      CONIN1: IN A,(KEYSTAT)   ;   キーボードから1文字入力
01BF-F9BF  CB47              BIT 0,A          ;
01C1-F9C1  20FA              JR NZ,CONIN1     ; もしA(0)=0ならCONIN1へ
01C3-F9C3  DB7C      KEY2:   IN A,(KEY)       ; ポートAから入力
01C5-F9C5  00                NOP              ;
01C6-F9C6  E67F              AND 01111111B    ;
01C8-F9C8  47                LD B,A           ;
01C9-F9C9  DB7E      CONIN3: IN A,(KEYSTAT)   ;
01CB-F9CB  CB47              BIT 0,A          ;
01CD-F9CD  28FA              JR Z,CONIN3      ;
01CF-F9CF  78                LD A,B           ;
01D0-F9D0  C1                POP BC           ;
01D1-F9D1  C9                RET
01D2-F9D2  CDBCF9    CONIN2: CALL CONIN       ; サブルーチン　CONIN2
01D5-F9D5  CD23FA            CALL CONOT       ;   1文字キー入力後即表示
01D8-F9D8  C9                RET
01D9-F9D9  3E00      CRTC: LD A,0             ; サブルーチン　CRTC
01DB-F9DB  47                LD B,A           ;   HD46505初期設定
01DC-F9DC  DD21F0F9          LD IX,CRPR0      ; パラメータテーブルアドレスセット
01E0-F9E0  D3F6      CRTC1: OUT (0F6H),A      ; レジスタセット
01E2-F9E2  DD7E00            LD A,(IX+0)      ;
01E5-F9E5  D3F7              OUT (0F7H),A     ; レジスタライト
01E7-F9E7  DD23              INC IX           ;
01E9-F9E9  04                INC B            ;
01EA-F9EA  78                LD A,B           ;
01EB-F9EB  FE0E              CP 14            ; 14回繰り返す
01ED-F9ED  38F1              JR C,CRTC1       ;
01EF-F9EF  C9                RET                HD46505パラメータテーブル
01F0-F9F0  3F        CRPR0:  DEFB  63         ; 水平総文字数
01F1-F9F1  28        CRPR1:  DEFB  40         ; 水平表示文字数
01F2-F9F2  32        CRPR2:  DEFB  50         ; HSYNC
01F3-F9F3  04        CRPR3:  DEFB   4         ; パルス幅
01F4-F9F4  19        CRPR4:  DEFB  25         ; 垂直総文字数
01F5-F9F5  01        CRPR5:  DEFB   1         ; 垂直ラスタ調整
01F6-F9F6  14        CRPR:   DEFB  20         ; 垂直表示文字数
01F7-F9F7  16        CRPR7:  DEFB  22         ; VSYNC
01F8-F9F8  00        CRPR8:  DEFB   0         ; ノンインタレースモード
01F9-F9F9  09        CRPR9:  DEFB   9         ; 最大ラスタアドレス
01FA-F9FA  65        CRPR10: DEFB  65H        ; カーソルスタートアドレス
01FB-F9FB  08        CRPR11: DEFB   8         ; カーソルエンドアドレス
01FC-F9FC  00        CRPR12: DEFB   0         ; スタートアドレス(H)
01FD-F9FD  00        CRPR13: DEFB   0         ; スタートアドレス(L)
01FE-F9FE  E5        CREA: PUSH HL            ; サブルーチン　CREA
01FF-F9FF  2100F0            LD HL,VSTAT      ;   全画面クリア
0202-FA02  3E20      CREA2: LD.A,' ' ;          VRAMアドレスF000~F3FF
0204-FA04  77                LD (HL),A
0205-FA05  7C                LD A,H           ;
0206-FA06  23                INC HL           ;
```

図 7.19　モニタプログラムのリスト(4)

```
0207-FA07  FEF4               CP 0F4H            ;
0209-FA09  20F7               JR NZ,CREA2        ;
020B-FA0B  E1                 POP HL             ;
020C-FA0C  C9                 RET
--------------------------------------------------------------------------
020D-FA0D  7E        PMSG:  LD A,(HL) ;          サブルーチン　PMSG
020E-FA0E  B7                 OR A               ;  メッセージ出力
020F-FA0F  C8                 RET Z              ; 0がくるまで繰返し表示する
0210-FA10  4F                 LD C,A             ;
0211-FA11  CD23FA             CALL CONOT         ;
0214-FA14  23                 INC HL             ;
0215-FA15  18F6               JR PMSG            ;
--------------------------------------------------------------------------
0217-FA17  E5        CRLF:  PUSH HL              ; サブルーチン　CRLF
0218-FA18  2120FA             LD HL,MCRLF        ;  復改および改行を行なう
021B-FA1B  CD0DFA             CALL PMSG          ;
021E-FA1E  E1                 POP HL             ;
021F-FA1F  C9                 RET
--------------------------------------------------------------------------
0220-FA20  0A0D00    MCRLF:  DEFB LF,CR,0        ; サブルーチン　CONOT
0223-FA23  F5        CONOT:  PUSH AF             ;  1文字ディスプレイ表示
0224-FA24  C5                 PUSH BC            ;
0225-FA25  D5                 PUSH DE            ;
0226-FA26  E5                 PUSH HL            ;
0227-FA27  3201F4             LD (BUFF),A        ;
022A-FA2A  FE0D               CP CR              ;}
022C-FA2C  2838               JR Z,CARET         ;} CRならCARETへ
022E-FA2E  FE0A               CP LF              ;}
0230-FA30  2841               JR Z,LFEED         ;} LFならLFEEDへ
0232-FA32  FE7F               CP RUB             ;}
0234-FA34  2814               JR Z,DELETE        ;} DELETEならDELETEへ
0236-FA36  CDBBFA             CALL PDISP         ; 1文字表示
0239-FA39  3A00F4             LD A,(HCOUNT)      ;
023C-FA3C  FE27               CP 39              ;
023E-FA3E  CAABFA             JP Z,DISP2         ; 40文字目が表示されたらDISP2へ
0241-FA41  3C                 INC A              ;
0242-FA42  3200F4             LD (HCOUNT),A      ;
0245-FA45  E1        DISPE:  POP HL              ;
0246-FA46  D1                 POP DE             ;
0247-FA47  C1                 POP BC             ;
0248-FA48  F1                 POP AF             ;
0249-FA49  C9                 RET
..........................................................................
024A-FA4A  3A00F4    DELETE: LD A,(HCOUNT)       ; DELETEコードの処理
024D-FA4D  B7                 OR A               ;
024E-FA4E  2816               JR Z,CARET
0250-FA50  3A00F4             LD A,(HCOUNT)      ;
0253-FA53  3D                 DEC A              ;
0254-FA54  3200F4             LD (HCOUNT),A      ;
0257-FA57  3E20               LD A,' '           ;
0259-FA59  3201F4             LD (BUFF),A        ;
025C-FA5C  CDBBFA             CALL PDISP         ;
025F-FA5F  2B                 DEC HL             ;
0260-FA60  CDE2FA             CALL CARSOL        ;
0263-FA63  C345FA             JP DISPE
..........................................................................
0266-FA66  3E00      CARET:  LD A,0              ; CRの処理
0268-FA68  3200F4             LD (HCOUNT),A      ;
026B-FA6B  21F8F2             LD HL,VMLD         ;
026E-FA6E  CDE2FA             CALL CARSOL        ;
0271-FA71  18D2               JR DISPE           ;
0273-FA73  CD78FA    LFEED:  CALL LSCROL         ; LFの処理
0276-FA76  18CD               JR DISPE
--------------------------------------------------------------------------
0278-FA78  DD2128F0  LSCROL: LD IX,VSTAT+1*LINE  ; サブルーチン　LSCROL
027C-FA7C  2100F0             LD HL,VSTAT        ;   1行分スクロール
027F-FA7F  DD7E00    LSCR1:  LD A,(IX+0)         ;
0282-FA82  CDD3FA             CALL DBLNK         ; ディスプレイブランキング時間にのみVRAMアクセス
0285-FA85  77                 LD (HL),A          ;
0286-FA86  23                 INC HL             ;
0287-FA87  DD23               INC IX
0289-FA89  7C                 LD A,H             ;
028A-FA8A  FEF3               CP 0F3H            ;
028C-FA8C  20F1               JR NZ,LSCR1        ;
028E-FA8E  7D                 LD A,L             ;
028F-FA8F  FE40               CP 40H             ;
0291-FA91  20EC               JR NZ,LSCR1        ;
0293-FA93  DB7E               IN A,(7EH)         ; キーボードストローブ入力チェック
0295-FA95  CB47               BIT 0,A            ;
0297-FA97  200C               JR NZ,LSCR2        ; 入力無の場合はLSCR2へ
0299-FA99  DB7C               IN A,(7CH)         ;
```

図7.19　モニタプログラムのリスト(5)

```
029B- FA9B  2F                  CPL                 ;
029C- FA9C  E67F                AND 01111111B       ; コントロールSが入力されたか？
029E- FA9E  FE13                CP CTRLS            ;
02A0- FAA0  2003                JR NZ,LSCR2           コントロールSでなければLSCR2へ
02A2- FAA2  CDBCF9      CONTS:  CALL CONIN          ; 再スタートのためのキー入力待ち
02A5- FAA5  1620        LSCR2:  LD D,32             ;
02A7- FAA7  CDFFFA              CALL TIME           ; 表示を見やすくするための無駄時間32×2msec
02AA- FAAA  C9                  RET
02AB- FAAB  CD78FA      DISP2:  CALL LSCROL         ; 40文字をこえた場合の処理
02AE- FAAE  3E00                LD A,0              ;
02B0- FAB0  3200F4              LD (HCOUNT),A       ;
02B3- FAB3  21F8F2              LD HL,VMLD          ;
02B6- FAB6  CDE2FA              CALL CARSOL         ;
02B9- FAB9  188A                JR DISPE            ;
02BB- FABB  C5          PDISP:  PUSH BC             ; サブルーチン `PDISP
02BC- FABC  21F8F2              LD HL,VMLD          ;   1文字表示
02BF- FABF  3A00F4              LD A,(HCOUNT)       ;
02C2- FAC2  4F                  LD C,A              ;
02C3- FAC3  0600                LD B,0              ;
02C5- FAC5  09                  ADD HL,BC           ;
02C6- FAC6  3A01F4              LD A,(BUFF)         ;
02C9- FAC9  CDD3FA              CALL DBLNK          ;
02CC- FACC  77                  LD (HL),A           ;
02CD- FACD  23                  INC HL              ;
02CE- FACE  CDE2FA              CALL CARSOL         ;
02D1- FAD1  C1                  POP BC              ;
02D2- FAD2  C9                  RET
02D3- FAD3  F5          DBLNK:  PUSH AF             ; サブルーチン DBLNK
02D4- FAD4  DB7E        DBLNKH: IN A,(DPSTAT)       ;   ディスプレイブランキング時間の検査
02D6- FAD6  CB57                BIT 2,A             ;   Cポートの2ビット目を使用
02D8- FAD8  28FA                JR Z,DBLNKH         ;   HighからLowになるのを待つ
02DA- FADA  DB7E        DBLNKL: IN A,(DPSTAT)       ;
02DC- FADC  CB57                BIT 2,A             ;
02DE- FADE  20FA                JR NZ,DBLNKL        ;
02E0- FAE0  F1                  POP AF              ;
02E1- FAE1  C9                  RET
02E2- FAE2  3E0E        CARSOL: LD A,14             ; サブルーチン CARSOL
02E4- FAE4  CDD3FA              CALL DBLNK          ;   HLレジスタの番地にカーソルディスプレイ表示
02E7- FAE7  D3F6                OUT (0F6H),A        ;
02E9- FAE9  7C                  LD A,H              ;
02EA- FAEA  E60F                AND 00001111B       ;
02EC- FAEC  CDD3FA              CALL DBLNK          ;
02EF- FAEF  D3F7                OUT (0F7H),A        ;
02F1- FAF1  3E0F                LD A,15             ;
02F3- FAF3  CDD3FA              CALL DBLNK          ;
02F6- FAF6  D3F6                OUT (0F6H),A        ;
02F8- FAF8  7D                  LD A,L              ;
02F9- FAF9  CDD3FA              CALL DBLNK          ;
02FC- FAFC  D3F7                OUT (0F7H),A        ;
02FE- FAFE  C9                  RET
02FF- FAFF  C5          TIME:   PUSH BC             ; サブルーチン TIME
0300- FB00  F5                  PUSH AF             ;   DレジスタがFFHで約0.5sec
0301- FB01  010000              LD BC,0             ;
0304- FB04  03          TIME2:  INC BC              ;
0305- FB05  78                  LD A,B              ;
0306- FB06  BA                  CP D                ;
0307- FB07  20FB                JR NZ,TIME2         ;
0309- FB09  F1                  POP AF              ;
030A- FB0A  C1                  POP BC
030B- FB0B  C9                  RET
```

図 7.19 モニタプログラムのリスト(6)

■**バンク切換え**

このモニタプログラムでは，全領域をRAM領域とするバンク切換えを行なっています。この切換えによりEPROMが切り離され，64 Kバイトの全メモリアドレス空間に対してREAD／WRITEが可能となります。

バンク切換えのプログラムは，図 7.19 に示したモニタプログラムリストの先頭部分にあります。これは，**図 7.21** に示すようにEPROMに書き込まれている全プログラムを，RAMのアドレスF 800 HからFFFFHへ移動させ，次に移動したRAMプログラム領域へJUMP（EH番地のJUMP　F 811 命令）し，JUMPしたRAMの最初の作業により， 0 番地から 07 FFH番地までのアドレス空間を占めていたEPROM領域を切り離すという手順を用いています。図 7.19 では

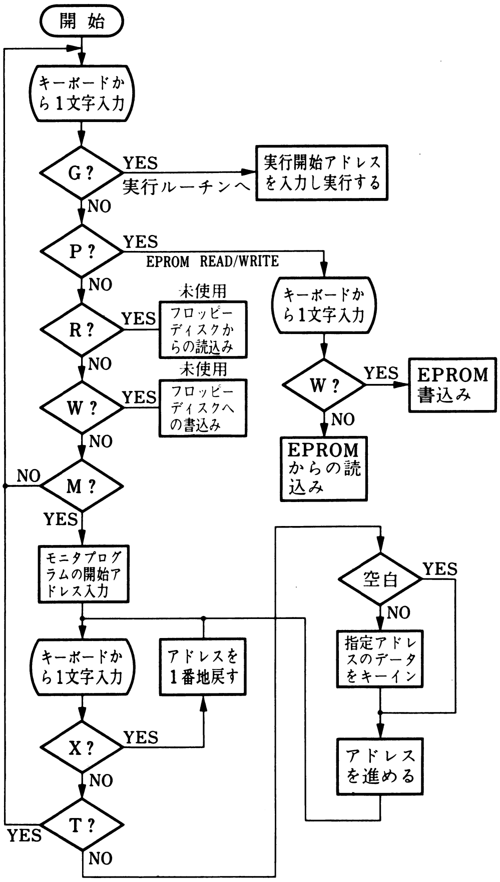


図 7.20　モニタプログラムのフローチャート

EPROMに書き込まれる番地と，バンク切換え後の番地を併記しました。以下，図8.18，図9.11，図9.15も同様です。

ハードウエア的には，図4.8に示したキーボード制御用のインターフェースIC　8255 AのCポートのPC_4，PC_5を用いて切換え操作を実行しています。このように，図7.19に示したプログラムリストのバンク切換えを行なった後の“F 811 H”番地以降は，実際にはROMに書かれた番地数よりF 800番地だけ多い番地に移動して記憶されていることになります。

■バンク切換えのテスト

モニタプログラムをROMに書き込んだら，CPU基板に挿入し，RESETスイッチを押します。そして，CRTディスプレイ画面左下隅に図

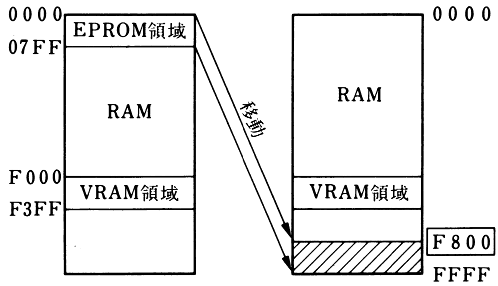


図 7.21　EPROM領域のプログラム移動

表 7.1　モニタプログラムのコマンド

コマンド名	機　能	使　用　法
M	メモリのREAD/WRITE	16進4桁でアドレスをキーインすることにより，メモリの内容を表示， この状態で2桁の16進をキーインすればメモリ内容を新しく書き込み， “空白”をキーインすればアドレスだけ進む Tを押せばモニタの初期状態へ復帰，また1番地分のアドレスのもどしは“X”
P W	PROM ライタ	1000番地から17F Fまでのデータを EPROM に自動書込み
P R	PROM リード	EPROM の内容を読み込み，1000番地から17F F番地までにストア
G	実　行	16進4桁で開始番地をキーインすることにより，その番地から実行する
コントロールS	ディスプレイ画面を一時停止する	
W	フロッピーディスクへの書込み	CP/M モニタで使用第8章参照
R	フロッピーディスクからの読込み	〃

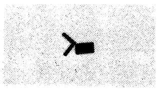


図 7.22 モニタプログラムの初期状態

7.22 に示すような“＞”が表示されれば，正しくバンク切換えが行なわれたことになります。

もし“＞”が表示されなければ，ハードのエラーとしてバンク切換え不良と考えられます。正常なら図 4.8 のキーボード基板上のCポートのPC_4 (13 番ピン) が“Low”に，またPC_5 (12 番ピン) が“High”になっていて，図 4.1 のCPU基板図上の**4 F**に示した$\overline{\text{BANK SELECT}}$信号が“Low”となっているはずです。したがって，$\overline{\text{BANK SELECT}}$信号が“Low”にならない場合は，ふたたびキーボード基板上のインターフェースIC 8255 Aのテストプログラム（図 7.7）によって，8255 Aの動作を確かめて下さい。

$\overline{\text{BANK SELECT}}$信号が“Low”になっていても，“＞”が表示されない場合は，プログラムの書込みミス，またはCRTコントローラの不良が原因と考えられます。手動のPROMライタによって，書込みプログラムの再チェックを行ない，またCRTコントロールテストプログラム（図 7.10）でCRTが正しい動作をするか検査して下さい。

なお，バンク切換えが正しく行なわれていることの確認として，本来ROM領域であった 0000 H～07 FFH番地に，“M”モードで任意のデータを書き込み，ふたたびその内容を読み出して正しく動作するかを確かめて下さい。

■モニタプログラムの使用法

モニタプログラムを書き込んだEPROMをCPU基板に挿入し，RESETスイッチを押すと，画面左隅に“＞”とカーソルが表示されます（図 7.22 参照)。これは，モニタのコマンド要求指令で，プログラムが動作状態になっていることを示しています。

この状態で“M”を押すと，“ADR”と表示され，4 桁のアドレスキーイン指令が表示されます。このとき，4 桁の 16 進数を入力すると，**図 7.23** に示すように，アドレスの内容が表示されます。

この後は，2 桁の 16 進を入力するごとに，カーソルの指示するアドレスの内容が変化します。もし誤ってキーインした場合などは，外付けのリセットスイッチを押すと，モニタの初期状態へもどるので，ふたたび“M”とアドレスをキーインすれば，まちがえた番地の内容を訂正できます。

```
 0FFB FB
 0FFC 00
 0FFD FF
 0FFE 00
 0FFF 00
■1000 FF
 1001 00
 1002 FF
 1003 00
 1004 FF
 1005 00
 1006 FF
 1007 00
 1008 FF
 1009 00
 100A FF
 100B 00
 100C FF
 100D 00
```

図 7.23 メモリ内容の表示例

なお，メモリの内容を変えずにアドレスだけを進めたいときは，“スペース”を打てばいいようになっています。また，誤ってキーインした場合に“X”を押すと 1 つ前の番地が表示されます。さらに，16 進数キーインに対する簡単なチェックとして，0 のアスキーコードの 30 Hより小さいコードと下のアスキーコードの 47 Hより大きなコードが入力された場合は，“？”が表示され，正しい英数字の 0 ～ 9 またはA～Fのキーイン待ちとなります。

■未使用の“R”と“W”の処理

図 7.19 のプログラムリストでは，“W”の飛び先番地はFB 0 CH番地となっていますが，バンク切換えによりF 800 H番地分だけ多い番地に配置されていることになっています。したがって，ROM上では 30 CH番地が対応します。そこで，誤ってWをキーインしてもすぐにモニタに戻るように，30 CH番地には，C 9 H（リターン命令）を書き込んで下さい。同様に，“R”の飛び先番地FB 4 AH，すなわち 34 AH番地にもC 9 Hを書き込みます。

■“G”コマンド

“G”コマンドを用いれば，16 進 4 桁で実行開始番地をキーインすることにより，プログラムの動作確認が行なえます。この“G”コマンドが正しく実行されたことを，任意の番地に 7.1 節に示した各種基本回路のテストプログラムをキーインして，実行させてることにより確認してみて下さい。

■PROMライタ

プログラムをEPROMに書き込む作業は，1000番地から機械語を順次打ち込み，作業終了後，“PW”をキーインすることにより，自動的にEPROMに書き込まれることになります。この書込み作業中は，現在書込みが行なわれているアドレスとそのデータがCRTに表示され，同様にEPROM上のアドレスおよびデータ表示用のLEDが点灯し，書込みの進行状況がわかるようになっています。

途中で打ち切りたいときは“RESET”スイッチを押すことにより，モニタの初期状態へ復帰し，強制的にROM書込み作業が中断します。

■EPROMリード

EPROMの内容を読み込み，1000番地からストアする作業は“PR”とキーインすることにより行なえます。EPROMの読込み作業が行なえない場合は，次に示すハード上のチェックを行なって下さい。

① 図6.5に示したEPROMライタ配線図中の**3C**にあるEXOR（74 LS 86）の入力が2本とも“High”となり，その出力が“Low”で$\overline{G}$が“Low”となっていること

② READ作業中は，同図**5D**のi 2716の$\overline{CS}$およびPD／PGMが“Low”状態を保持していること

③ また同図**4C**の74 LS 245の1番ピン“DIR”が“Low”状態を保ち，データはBからAへつつ抜けになっていること

以上の点が正しく動作していれば，指定アドレスの内容は必ずREADされていることになります。**表7.2**に，EPROMへのREAD／WRITE作業時における信号の様子を示します。

表7.2　PROMライタ回路の信号の様子

	READ作業	WRITE作業
チップセレクト $\overline{CS}$	“Low” 一定	クロック
プログラムモード信号PD/PGM	“Low” 一定	クロック
LS245出力コントロール$\overline{G}$	“Low” 一定	PD/PGMの逆位相信号
LS245方向制御DIR	“Low” 一定	“High” 一定
＋25V電源スイッチ	OFF	ON

《演習問題》

現在CPU基板上に挿入されていて，モニタとして動作しているEPROMと同じ内容でコピー用のEPROMを作りなさい。

（解法） 2716と同じ内容のプログラムは，F800H番地以降に配置されていますので，その内容を1000H番地以後に移し，“PW”コマンドを用います。また，“PW”の開始番地をモニタによりF800Hに変更すれば，この訂正した番地以外の部分のコピーが作られます。

■ディスプレイブランキング

ラスタスキャンディスプレイは，独自にVRAMアクセスを繰り返してディスプレイ画像を発生させています。VRAMに対するデータの書換え作業をCPUから任意の時刻に行なうと，CPUからの書込みデータがキャラクタジェネレータを通してディスプレイされるため，画面がちらついて見苦しくなります。

この画面の乱れを防止するには，各種の方式があります。[7-1]しかし，ここで製作するシステムではハードウエア的にもっとも簡単な，ディスプレイブランキング時間を利用して，VRAMへのデータの書込みおよびHD 46505の各レジスタへの書込み作業を行なう方式を採用することにします。

HD 46505では，水平復帰区間に**図7.24**に示すようなディスプレイブランキング信号である“DISTING”信号が発生します。この信号を

キーボード制御用の8255のCポートの入力PC_2を介して，ブランキング時間として検出しています。

このようなブランキング時間を利用する方法は，ディスプレイ表示に時間がかかることが難点といわれています。ここで作成したモニタでは，これでも速すぎて，無駄時間を挿入しているくらいなので，この方式が最適と考えられます。

なお，ブランキングの検出部は，モニタのプログラムリスト中のサブルーチンDBLNKで行なっており，ほぼ完全に乱れないように**図 7.25**に示すDISTINGが"High"から"Low"に変化した時点だけで，ストア命令を実行しています。

■コントロール"S"の使用

自動によるROMライタ作業において，ディスプレイ画面を一時停止して内容を確認したい場合があります。このようなときのディスプレイ画面の一時停止は，コントロール"S"をキーインすることによって行ないます。

コントロールSとは，キーボード上に配置されている"コントロールキー"と"S"を同時に押すことです。ハードウエア的には，1バイトのアスキーコードの第6ビット目が"0"となることで，**図 7.26**に示すように"S"の53Hが13Hとなります。

もしコントロールキーがついていないキーボードを使用する場合は，コントロール"S"でなくとも，自分で決めた任意のコードでこの作業が可能です。この場合は，モニタリスト中のFA9EH番地の，

FA9E FE 13 CP CTRLS

を，CP ××（指定するコード）に変更して下さい。

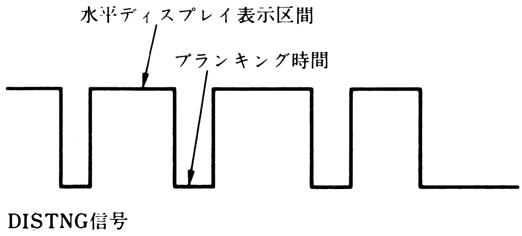


図 7.24 HD 46505のディスプレイタイミング信号の様子

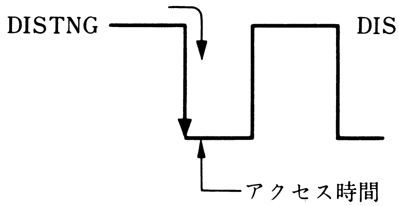


図 7.25 DISTINGの"High"から"Low"への移行時点の検出

	ビット 7 6 5 4 3 2 1 0
アスキーコードS (53H)	0 1 0 1 0 0 1 1
コントロールS (13H)	0 0 0 1 0 0 1 1
	↑ 第6ビット目が"0"となる

図 7.26 コントロールキーを押すことによるコードの変化

《雑音対策あれこれ》

通常は正しく動作するが，ときどき誤った動作をするとか，ゆらしたり動かしたりすると調子が悪くなる，長時間使用すると調子がおかしくなる……などの経験をもっている人が多いと思います。このような装置は，信頼性の低い装置と呼ばれ，より信頼性の高い装置を作ることが必要となります。

どのようにすれば信頼性の高い装置が作り出せるかということに対する答えは，“経験をつむことである”といえます。誤動作に対し，あれこれ対策し，悩み，そして解決したその経験こそが信頼性の高い装置を作り出すための最良の学習です。

誤動作の原因の大半は，イモハンダ，ソケットの接触不良，コードの断線寸前状態などの実装技術に関することに起因しています。この場合は，信頼性の高い部品を使用して，ていねいに配線を行なうことが重要です。

その他の原因としては，雑音が考えられ，この雑音対策は大変やっかいな問題です。主な雑音対策を列挙してみます。

① 外部装置との接続

雑音が混入する原因としては，外部装置との接続のときに生じるものがあります。外部装置とのケーブルは，できるだけ短く，かつツイストペアとする方がいいでしょう。

② バスラインの強化

誤動作の原因としては，各バスラインの入出力能力に起用するものが考えられます。例えば，キーボードインターフェースとPROMライタ回路の8255 Aは，バスバッファを通さず直接データバスに接続しています。この状態を考えてみましょう。

8255 Aの出力は，“Low”状態で電圧が0.45 Vのとき1.7 mAのシンク能力があります。一方，データバスに接続されるICは，CPU，FDCの2個の74 LS 245です。この74 LS 245の“Low”のときのソース電流は，0.4 mAなので，2個で0.8 mAの電流が8255 Aに流れ込むことになります。したがって，直流条件だけを考えた場合は8255 Aの出力で十分バスのドライブができると考えられます。

一方，交流条件すなわち過渡状態を考えると，メモリの4164 1個とコネクタ間を結ぶバスの配線のもつ容量C（キャパシタンス）が問題になります。なぜなら，Cがあると波形がなまり，この波形のなまりは信号源インピーダンスが高いほど大きくなります。

したがって，駆動能力の大きな素子でバスをドライブするほど，高速で動作させることができます。また，外部からの雑音に対しても低いインピーダンスで駆動するほど，耐性が増加します。雑音原因となるものが周囲に多い環境で使用する場合や，誤動作が多い場合には，バッファ（例えば74 LS 245）を挿入する必要があります。この考え方は，EPROM2716にも適用されます。

③ アース対策

外部装置と本体は，太いアース線で接続するなどのアース対策が必要です。

④ 雑音源の除去

その他，雑音源となるものは極力近くに置かないということが必要です。とくにスイッチング電源は強力な雑音源ですので，フロッピーディスク装置などからできるだけ離した場所に設置して下さい。

《熱対策》

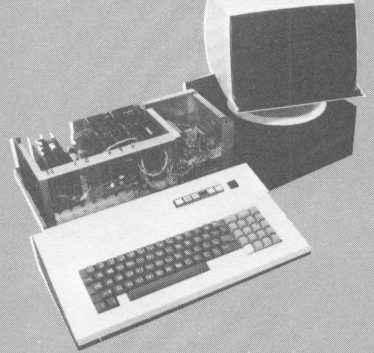
長時間の使用により，トランスや3端子レギュレータ，スイッチング電源などが触れられないぐらい熱くなることがあります。このようなときには，必ずファンを取り付けて放熱して下さい。

第1部
CP/Mマイコンのハード作製

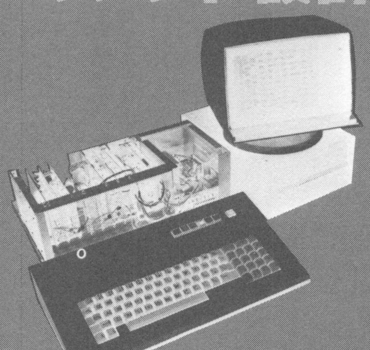
第2部
CP/Mマイコンのソフト設計

第3部　付　録

第1部　CP/Mマイコンのハード作製
第2部　CP/Mマイコンのソフト設計
第3部　付　録



第1部　CP/Mマイコンのハード作製
第2部　CP/Mマイコンのソフト設計
第3部　付　録



第2部第8章

フロッピーディスク制御回路の作り方

8

手作りコンピュータを有効に利用し，市販されているマイコンなみの働きをさせるために，標準フロッピーディスク装置を装備させて，CP/M が動作するようにします。ここでは，片面単密度の標準フロッピーディスク装置の制御回路の作り方を述べます。

8.1 フロッピーディスクについて

フロッピーディスクは，薄型，軽量で，取扱いやすく，また低価格であるという特徴があるため，外部記憶装置として大変普及してきました。このフロッピーディスクには，いろいろな仕様のものがあり，**表8.1**のようになっています。

フロッピーディスクには，直径8インチと5.25インチの2種類のものがあり，8インチフロッピーディスクを標準ディスク，5.25インチフロッピーディスクをミニフロッピーとよんでいます。また，記憶密度は，単密度と倍密度の2種類，記憶面は片面と両面の2種類があります。片面単密度に対し，両面倍密度は4倍の記録容量をもつことになります。

ミニフロッピーディスクは，現在規格化が進められ，統一化の方向に向かっています。とくに，8インチフロッピーディスクは国際標準規格化が実現され，記録媒体の互換性があるため，各種システム間でのデータ交換に広く利用されています。

フロッピーディスクに関しては，電子科学ブルーブックス①（産報出版）に“実戦的フロッピーディスク装置入門”という参考書がありますので，詳しく知りたい方はそちらを参照して下さい。本節では，フロッピーディスク制御回路設計に必要な事項についてのみ述べます。

■標準フロッピーディスク

片面標準フロッピーディスクは，**図8.1**に示すように，00から76までのトラックとよばれる77本の円周上の記録領域から構成されます。また両面フロッピーディスクも，同様のトラック構成となっていて，両面に対応する1対のトラックをシリンダとよんでいます。両面フロッピーディスクは，サイド0とサイド1にそれぞ

表8.1　フロッピーディスクの種類

	種類	記録容量（MB）	トラック数	セクタ数
8インチ	片面単密	0.243	77	24
	両面単密	0.493	154	
	両面倍密	0.985	154	
5インチ	片面単密	0.080	35	18
	両面倍密	0.286	70	16

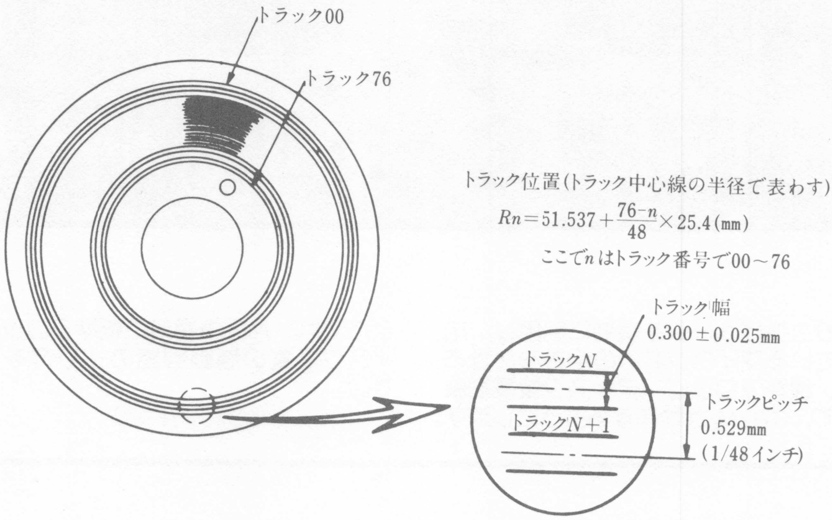


図 8.1　トラックの構成(1)

れ対応して，合計154本のトラックが存在することになります。

■**データフォーマット**

1本のトラックは，**図8.2**に示すデータフォーマットからなっており，データを記憶する26のセクタ部と，セクタの開始を示すプレアンブル部およびセクタの終わりを示すポストアンブル部があります。さらに，1つのセクタは**図8.3**に示すように，先頭にIDフィールドとよばれるセクタの開始とアドレスを示す情報を記録した部分と，128バイトのデータを記録するデータフィールドから構成されています。

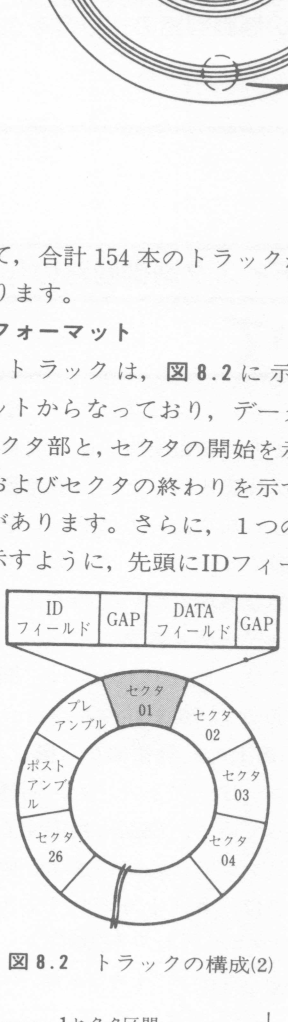


図 8.2　トラックの構成(2)

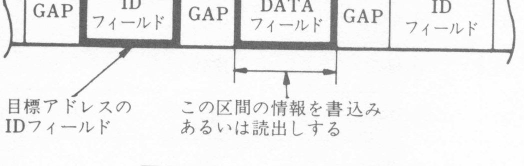


図 8.3　1セクタの構成

フロッピーディスクにデータを書き込むときは，指定したトラック上でIDフィールド情報を読み込み，セクタナンバーが合致したセクタに対して128バイトの記録データを書き込みます。同様にして，目標セクタと一致するIDフィールドをもったセクタからデータを読み出します。

このように，1本のトラックにはプレアンブル部やIDフィールドがあらかじめ書き込まれていなければ，読み書きができません。これらの必要な情報を，あらかじめフロッピーディスク上に書き込むことをイニシャライズまたはフォーマッティングとよんでいます。

■**フロッピーディスクを扱うときの注意**

フロッピーディスクは，きわめて簡単な構造になっていますから，取扱いには注意を必要とします。

①　大変軟かいものなので，曲げたり折ったりしないこと。とくにフロッピーディスクドライブ装置へ挿入するときには，まっすぐ静かに押し込み，乱暴に扱ってはなりません。また，使用しないときは，硬い表紙のファイルにとじておくか，収納箱に入れて垂直に立てかけて保管しておかなければなりません。

② **図8.4** に示すヘッドウインドウからディスクに指紋をつけないようにすることが大切です。フロッピーディスクを扱うときは，必ずラベルの部分をもち，使用しないときは1枚ずつカバーに入れておくことが重要です。

③ フロッピーディスクに磁石を近づけたり，鉛筆などの硬い筆記具でラベル書きをしてはいけません。

筆者らの経験では，フロッピーディスクはていねいに扱う限り，大変信頼性の高い記録装置であるといえます。

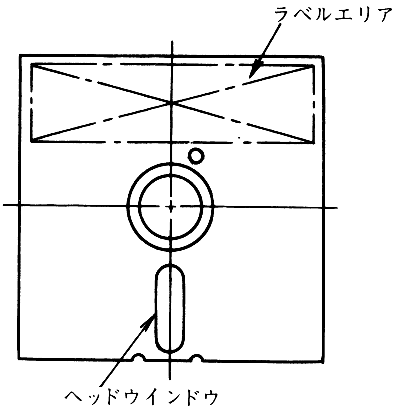


図8.4　標準ディスクの外観

8.2 フロッピーディスク制御回路の作り方

第8章　フロッピーディスク制御回路の作り方

ここでは標準フロッピーディスク装置として，YD-74 C（ワイ・イー・データ社製）を用いたフロッピーディスク制御基板の作り方について述べます。

この制御回路は，1台の標準フロッピーディスクの制御回路で，もっとも簡単な回路構成となっています。マイクロプロセッサとフロッピーディスクとのインターフェース信号の仕様や，フロッピーディスクインターフェースICの機能を理解するには，この制御回路の試作は最適な作業だと考えられます。

最近，両面倍密度のフロッピーディスク装置が安価になってきていますが，手作りマイコンの入出力装置としてはまだまだ高価です。世の中の流れは両面倍密度の時代に入りつつありますが，ここではフロッピーディスクの基本形である標準片面単密度ディスク装置のYD-74 Cを採用しました。

8.2.1 フロッピーディスクのコントロール

■フロッピーディスク装置

フロッピーディスク装置は，おもに次に示す機構から構成されています。

① 磁気ヘッド
② ヘッド位置決め機構
③ ヘッドロード機構
④ ディスク駆動機構
⑤ インデックス検出部

例をYD-74 Cにとると，基本ブロックは**図8.5**のようになります。各機構は次のとおりです。

磁気ヘッドは，フロッピーディスク面に接触し，データの読出しと書込みを行ないます。また，ヘッドの位置決め機構は，ステッピングモータとリードスクリューによって駆動され，指定トラックの位置決めが行なわれます。

ヘッドロード機構とは，読出し，書込み時のみに，ディスクとヘッドを接触させ，ヘッドの寿命を延ばすようにするためのソレイドと復帰スプリングから構成されています。

インデックス検出部は，ディスクにあけられたインデックスホールを，発光ダイオードとフォトトランジスタにより検出し，セクタの始まりを検出する機構です。

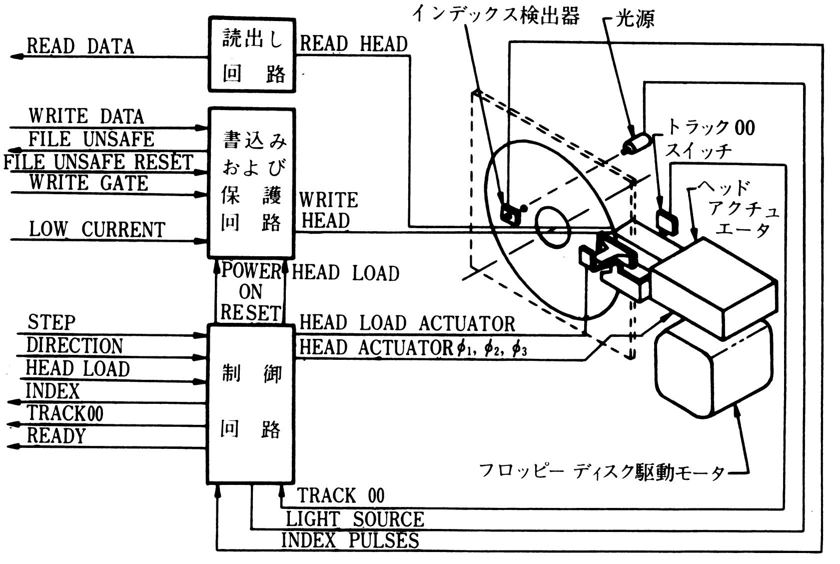


図8.5　YD-74 Cの基本ブロック図

■**YD-74 Cディスクドライブ**

YD-74 Cは，ワイ・イー・データ社製の8インチ片面単密度のフロッピーディスクドライブで，**表8.2**に示す機能をもっています。このYD-74 Cとディスクコントローラ間のインターフェース信号の種類は，**図8.6**に示す構成となっていて，各信号の機能は次のようになっています。

(1)　入力信号（インターフェースからYD-74 Cへの信号）

①　DIRECTION（31番ピン）
ヘッドの移動方向を決める信号
"High"アウター（トラック00）方向
"Low"インナー（トラック76）方向

②　STEP（29番ピン）
DIRECTION信号とともに使用し，1パルスで1トラック移動

表8.2　YD-74 Cの機能

メディア	トラック数	77
	セクタ数	26（IBMイニシャライズによる場合）
	トラック密度	48トラック／インチ
所要動作時間	ヘッドロードタイム ディスク回転数 アクセス時間	50msec 360RPM 最小6msec
記録	記録方式 データ転送速度	FM 250Kバイト／sec
最大消費電力	DC	65W（シーク動作時）
	AC	40W
定格	100V	1.5A
	＋5V	1.0A
	－12V	100mA
	24V	1.5A

③ HEAD LOAD（33番ピン）

"Low"によりヘッドとディスクが接触し，READ／WRITE可能となる。READ／WRITE動作に先立ち，少なくとも50 ms前に"Low"レベルにする必要がある

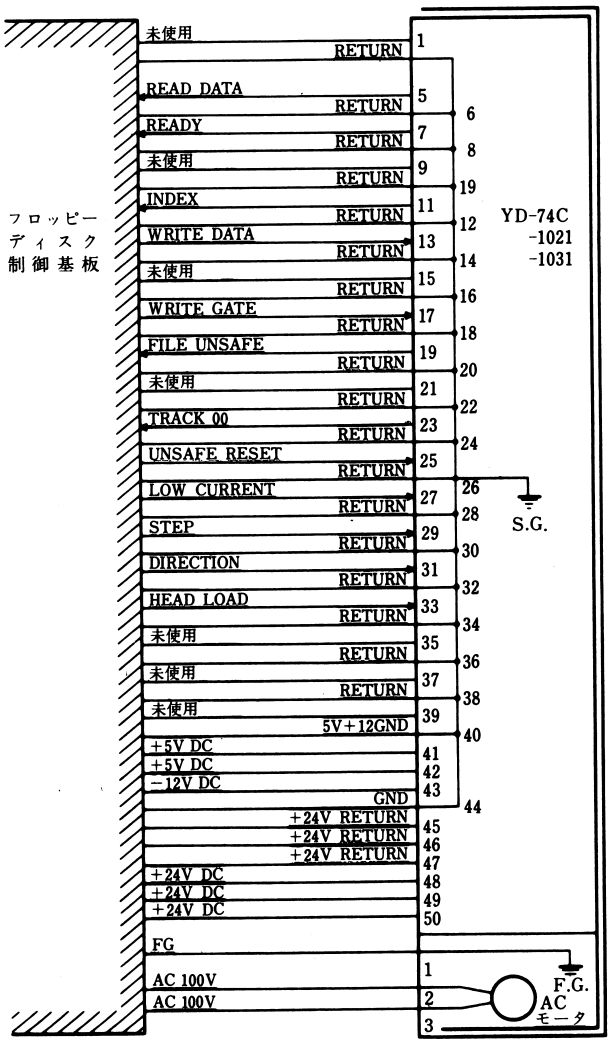


図 8.6 YD-74 Cのインターフェース信号

④　WRITE　GATE（17番ピン）

“Low”でディスクへのWRITEが可能となる

⑤　WRITE　DATA（13番ピン）

WRITE GATEが“Low”で，この入力信号にパルス列を与えることにより，FM変調されたデータの書込みが行なわれる

⑥　LOW　CURRENT（27番ピン）

ディスク記録面の外周期と内周期で書込み電流の切換えを行なう。トラック0～43で“High”レベル，44～76で“Low”レベルに設定する

⑦　UNSAFE　RESET（25番端子）

ディスクへの書込みが不可能となった場合，“Low”パルスを与えることにより，ふたたびディスクへの書込みが可能となる

(2)　出力信号（YD-74Cからインターフェースへの信号）

①　READY（7番ピン）

“Low”状態でREAD／WRITE準備完了を示す

②　TRACK　00（23番ピン）

“Low”状態でトラック00の位置にヘッドがあることを示す

③　INDEX（11番ピン）

ディスクが1回転するごとに，1パルスを送信

④　READ　DATA（5番ピン）

ディスク上に書き込まれたデータを検出し，フロッピーディスクインターフェースに出力信号を与える

⑤　FILE　UNSAFE（19番ピン）

ハードウエア上の誤動作またはオペレータミスなどの誤動作を検出すると“Low”になり，ディスクへの書込みを禁止する

■フロッピーディスクコントローラFD 1771

フロッピーディスクコントローラは，マイクロプロセッサ（CPU）とフロッピーディスク装置のインターフェース用の制御回路で，データの読み書きを制御する機能をもっています。現在では，多くのフロッピーディスク装置コントロール専用のLSIが入手可能となっています。ここで取り上げるシステムでは，単密度のフロッピーディスク装置を用いて，ウェスタンディジタル社のFD 1771を使用したコントローラ回路を作製します。

FD 1771は単密度専用（FM記録のみ）です*。このピン配置を**図8.7**に示します。

FD 1771は40ピンのnチャネルMOS-LSIで，プログラマブルな制御ができます。内部は**図8.8**に示すブロック図の構成で，5つのレジスタが準備されています。これらのレジスタは，下位アドレスA_0とA_1の指定により選択され，FD 1771とZ-80 CPUとの間のデータ，コントロールワードおよびステータスの送受信が行なわれます。A_0とA_1のアドレス選択と，各レジスタの関係を**表8.3**に示します。各レジスタの機能は次の通りです。

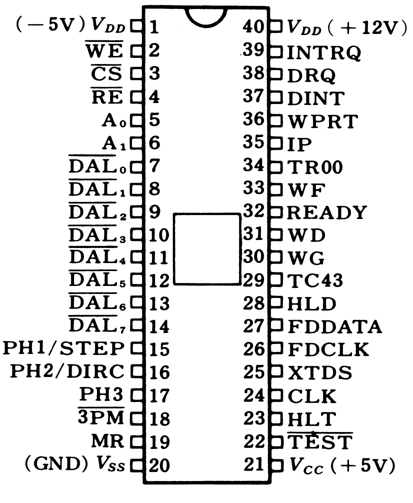


図8.7　FD 1771のピン配置

＊　現在，単密度，倍密度どちらにでも使用可能なインターフェースICが市販されています。例えば，MB 8877（富士通）やμPD-765（NEC）は，5V単一電源で作動し，かつ1つのチップで5インチ単密度から8インチ両面倍密ディスクまでに使うことができます。[8-2] FD 1771以外の専用ICの使用法については，ほかの文献を参照して下さい[8-1),8-2)]。

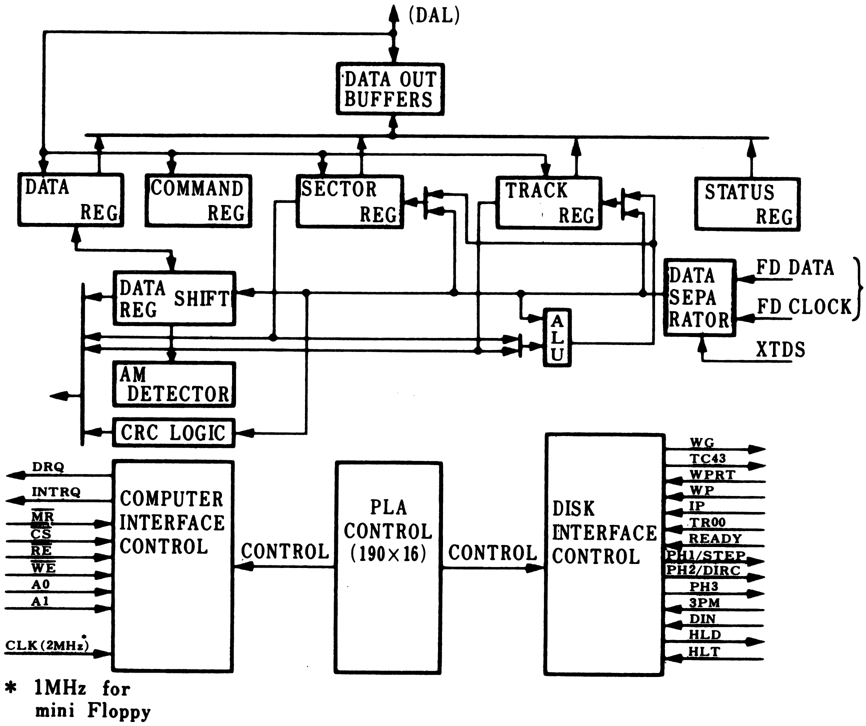


図 8.8 FD 1771 の内部構成

① ステータスレジスタ
フロッピーディスクの状態を保持し，DAL_0～DAL_7端子によりプロセッサへ知らせる

② トラックレジスタ
ヘッドがアクセス中のトラック番号を保持する。このレジスタの内容は，$DAL_{0\sim7}$端子からCPUへ読込みができる

③ セクタレジスタ
セクタアドレスを保持する。$DAL_{0\sim7}$端子から読込み可能

④ コマンドレジスタ
実行中のコマンドを保持する

⑤ データレジスタ
ディスクリードモードでは，このデータレジスタへ8ビットパラレルの信号がデータシフトレジスタから送られ，ディスクライトモードでは，データレジスタから8ビットパラレルの信号がデータシフトレジスタに送られる。シークコマンド実行中は，本データレジスタは指定トラックアドレスを保持する

表 8.3 FD 1771 のレジスタ

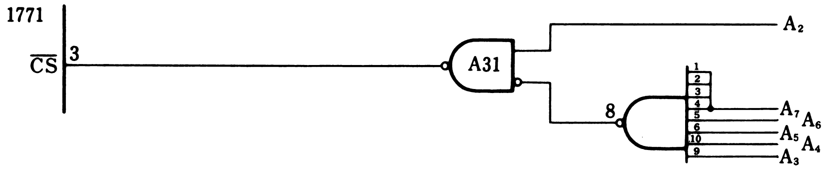
$A_1 - A_0$	READ (RE)	WRITE (WE)
0 0	Status Register	Command Register
0 1	Track Register	Track Register
1 0	Sector Register	Sector Register
1 1	Data Register	Data Register

8.2.2 フロッピーディスク制御基板の作り方

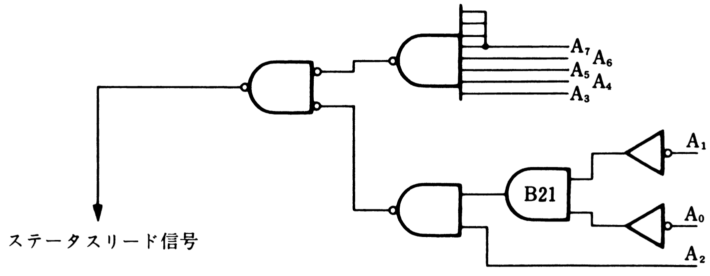
フロッピーディスク制御基板の配線図面を**図 8.9**(巻末折込みページ)に示します。Z-80 CPU と FD 1771 とのデータの送受信は，すべて 8 ビットデータバス ($DAL_{0\sim7}$) を通じて行なわれます。また，フロッピーディスクインターフェースの機器指定は，下位 8 ビットのアドレスを，

$$\begin{matrix} A_7 & A_6 & A_5 & A_4 & A_3 & A_2 & A_1 & A_0 \\ 1 & 1 & 1 & 1 & 0 & 1 & \times & \times \end{matrix}$$

と設定することによって行ないます。この指定を行なうのが同図中**6C**，**7C**にあるNAND回路です。**図 8.10**(a)に示すように，A_2～A_7がすべ



(a)　アドレスA_2～A_7の“High”によるチップセレクト信号の発生



(b)　A_0とA_1がともに“Low”の場合はステータスリード命令

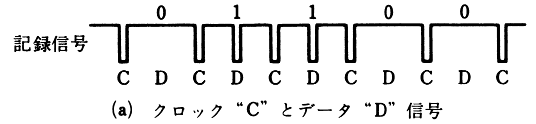
図8.10　図面中**6C**にある論理回路構成

て“High”のときチップセレクト信号$\overline{CS}$が“Low”となり，$DAL_{0\sim7}$とデータバスの入出力動作を行ないます。

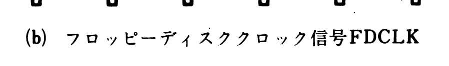
図8.10(b)に示す部分は，ステータスREAD動作（A_0とA_1が“Low”）のときに，READ／WRITE作業中（INTRQ）をCPU側へ送信するための回路です。

また，図8.9上の**5D**，**6D**，**7D**にある回路は，CPUのREADおよびWRITE動作に従い，データバスの方向を双方向バスコントローラ8T26を用いて制御する部分です。

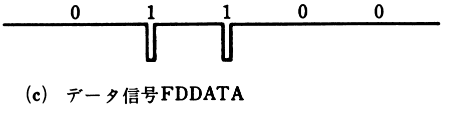
■データセパレータ回路



(a)　クロック“C”とデータ“D”信号



(b)　フロッピーディスククロック信号FDCLK



(c)　データ信号FDDATA

図8.11　ディスク記録信号からのクロックとデータの分離

図8.9上の**1C**，**2C**，**1D**，**2D**の回路は，データセパレータ回路です。ディスク上には，**図8.11**(a)に示すように，データビット“D”とクロックビット“C”が時系列信号として記録されています。

このディスク上のデータを読み取り（図8.9の**2B**のRDATA（5番ピン）から入力される），データセパレータ回路でデータビットとクロックビットに分離し，データビットをFDDATA端子(27番ピン)に，またクロックをFDCLK端子(26番ピン)に送ります。外部データセパレータを用いるので，$\overline{XTDS}$(25番ピン)を“Low”にします。

FD1771では，8インチフロッピーディスク装置を制御するには2MHz，ミニフロッピーディスクを制御するには1MHzを外部から与える必要があります。ここでは，CRTコントローラ用に作製した4MHz信号を利用し，LS175のD_3，Q_3，$\overline{Q}_3$を利用して2進のカウンタを構成し，2MHzにしてCLK端子(24番ピン)に入力します。

■YD-74Cの制御

図8.9の**2A**，**2B**に示す回路部分は，YD-74Cから送られた信号をFD1771が受信する部分で，同図**4A**，**4B**に示す部分がFD1771からYD

-74 Cへの信号です。これらのすべての入出力信号は，バッファを介して送受信を行なっています。

FD 1771からの制御信号のうち，STEP信号（15番ピン）に挿入されている単安定回路（74 LS 123）は，YD-74 Cステップ動作に必要な10 μsのパルス幅を発生させる回路です。さらに，ヘッドロードの時間として50 msが必要なので，HLD端子（28番ピン）の信号から単安定回路を介して，HLT端子（23番ピン）に接続しています。その他の信号の$\overline{\text{TEST}}$および$\overline{\text{3 PM}}$は＋5 Vに，$\overline{\text{DINT}}$端子は0 Vに固定しておきます。

8.2.3 フロッピーディスクコントロール基板のテスト

製作した基板の動作テストを行ないます。フロッピーディスクの各動作は，コマンドレジスタに対し，**表 8.4** に示す11種類のコマンドを書き込むことにより行なわれます。また**表 8.5** にステータスレジスタの意味について示します。これらのコマンドを使用して，フロッピーディスクが正しく動作するかどうか確認して下さい。

表 8.4 FD 1771のコマンド表

(a)

種別	コマンド名称	7 (MSB)	6	5	4	3	2	1	0 (LSB)
タイプI	リストア	0	0	0	0	h	V	r_1	r_0
	シーク	0	0	0	1	h	V	r_1	r_0
	ステップ	0	0	1	u	h	V	r_1	r_0
	ステップイン	0	1	0	u	h	V	r_1	r_0
	ステップアウト	0	1	1	u	h	V	r_1	r_0
タイプII	リードデータ	1	0	0	m	b	E	0	0
	ライトデータ	1	0	1	m	b	E	a_1	a_2
タイプIII	リードアドレス	1	1	0	0	0	1	0	0
	リードトラック	1	1	1	0	0	1	0	S
	ライトトラック	1	1	1	1	0	1	0	0
タイプIV	強制割込み	1	1	0	1	I_3	I_2	I_1	I_0

(b)

タイプ I
h＝ヘッドロードフラグ
h＝1　コマンド実行に先立ちヘッドロード
h＝0　コマンド実行に先立ちアンヘッドロード
V＝ベリファイフラグ
V＝1　最終トラックの照合を行なう
V＝0　照合を行なわない
$r_1 r_0$＝ステッピングモータレート
$r_1 r_0$　6 msec
0 0　6 msec
0 1　10msec
1 1　20msec

(c)

タイプ II
m＝マルチレコードフラグ
m＝0　シングルレコード
m＝1　マルチプルレコード
b＝ブロック長フラグ
b＝1　IBMフォーマット外（128～1024バイト）
b＝0　IBMフォーマット（16～4096バイト）
$a_1 a_2$＝データアドレスマーク
$a_1 a_2$
0 0　FB（データマーク）を書き込む
0 1　FA（ユーザ定義）を書き込む
1 0　F9（ユーザ未定義）を書き込む
1 1　F8（消去データマーク）を書き込む

(d)

タイプ III
S＝シンクロナイズフラグ
S＝0　アドレスマークに同期
S＝1　アドレスマークに同期しない

(e)

タイプ IV
I_i＝インタラプト条件フラグ
I_3のみ1　即，インタラプトに入る
I_2のみ1　インデックスパルス検出後インタラプト
I_1のみ1　Ready to Not Ready Transition
I_0のみ1　Not Ready to Ready Transition

表8.5　FD 771ステータス表

ステータスレジスタ						
ビット	全タイプI	リード・アドレス	リード	リードトラック	ライト	ライトトラック
7	Not Ready	Not Ready	Not Ready	Not Ready	Not Ready	Not Ready
6	Write Protect	0	Record Type	0	Write Protect	Write Protect
5	Head Engaged	0	Record Type	0	Write Fault	Write Fault
4	Seek Error	ID Not Found	Record Not Found	0	Record Not Found	0
3	CRC Error	CRC Error	CRC Error	0	CRC Error	0
2	Track 0	Lost Data	Lost Data	Lost Data	Lost Data	Lost Data
1	Index	DRQ	DRQ	DRQ	DRQ	DRQ
(LSB) 0	Busy	Busy	Busy	Busy	Busy	Busy

■フロッピーディスクリセット

はじめに,外部リセットスイッチによるフロッピーディスクのリセットテストを行ないます。フロッピーディスク装置内のステップモータの軸は,多少かたいですが手でまわすことができます(この際,精密機械ですので,無理をしないで下さい)。

したがって,ヘッドをディスクの中央くらいまで移動させ,この状態で外部リセットスイッチを押すと,ヘッドがホームポジション,すなわちトラック番号00の位置へ移動します。

ヘッドが移動しない場合は，以下の点をチェックして下さい。

① 基板端子A2の$\overline{\text{RESET}}$信号が正しくフロッピーディスク装置のRESET信号（25番ピン）に入力されているかどうか

② トラック番号00に移動するまでFD1771のSTEP（15番ピン）端子からパルスが発生され，フロッピーディスク装置入力の29番ピンに入力されているかどうか

③ FD1771のマスタリセット入力$\overline{\text{MR}}$（19番ピン）が“Low”となるかどうか

なお，トラック番号00に到達したとき，フロッピーディスク装置のTR 00（23番ピン）が“Low”となることを確認して下さい。

■基本動作テスト

モニタプログラムを利用して，指定番地から次に示す各基本テストプログラムをキーインし，開始番地から実行することによってフロッピーディスク装置の動作テストを行ないます。

(1) ステップインテスト

モニタにおける“M”コマンドで**図8.12**に示すステップインテストプログラムを3000番地から3006番地までキーインします。Tをキーインして，ふたたびモニタにもどし，実行コマンド“G”で3000番地からプログラムを実行すると，ヘッドが内側（トラック番号74の方向）へ1トラック分移動します。このプログラムリストの“LD　A,43 H”の43 Hが前にあげた表8.4(a)のコマンド表におけるステップイン命令に対応します。

(2) ステップアウトテスト

同様に**図8.13**に示すステップアウトテスト

```
3000  3E43     STEPIN  LD A,43H
3002  D3F8             OUT (0F8H),A
3004  C311F8           JP  MONITOR
```

図8.12　ステップインテストプログラム

```
4000  3E63     STEPOUT LD A,63H
4002  D3F8             OUT (0F8H),A
4004  C311F8           JP MONITOR
```

図8.13　ステップアウトテストプログラム

```
                       ORG 5000H
5000  3E8C     HLOAD   LD A,08CH
5002  D3F8             OUT (0F8H),A
5004  C311F8           JP MONITOR
```

図8.14　ヘッドロードプログラム

```
6000  3E09     SEEK    LD A,9
6002  D3FB             OUT (0FBH),A
6004  3E13             LD A,13H
6006  D3F8             OUT (0F8H),A
6008  C311F8           JP MONITOR
```

図8.15　シークテストプログラム

プログラムを4000番地からキーインして実行します。これによって，ヘッドが外側へ1トラック分移動します。

(3) ヘッドロードテスト

図8.14は，ヘッドロードテストプログラムです。このプログラムは，リードコマンドをヘッドロードあり，で実行させます。5000番地からプログラムを書き込んで実行し，ヘッドがディスクに接触すると“カタン”という音がするので，これで正しく実行されたことになります。

(4) シークテスト

シークテストプログラムを**図8.15**に示します。ここにあげたプログラム例では，6000番地に“LD A,9”と記述して，第9トラックへの移動を指定しています。

■リードテストプログラム

現在ヘッドが位置しているトラック上のセクタ番号1から1セクタ分(128バイト)のデータを読み込み，“5000 H”番地からストアするテストプログラムを**図8.16**に示します。このプログラムはモニタを用いてメモリの4000 H番地からキーインし，プログラムの実行も4000 H番地とします。

正常にリード作業が行なわれたら，モニタの先頭番地へＪＵＭＰします。また，読込み作業後の状態(ステータス；STATUS)を“4030 H”番地にストアします。このステータスの各ビットの意味は，前にあげた表8.5に示されており，“00 H”のとき完全にリード作業が行なわれたことを示しています。

```
LOC.  OBJ                    SOURCE STATEMENT

F811           MONITOR  EQU   0F811H
               ;
                        ORG 4000H
4000  210050   RSECTOR  LD HL,5000H       ; リードデータストア番地の設定
4003  3E01              LD A,1            ; セクタ番号1の設定
4005  D3FA              OUT (0FAH),A      ; セクタレジスタ書込み
4007  3E8C              LD A,8CH          ; リードコマンド設定
4009  D3F8              OUT (0F8H),A      ; リード開始
400B  DBFC     RLOOP    IN A,(0FCH)       ; INTRQに対するWAIT
400D  B7                OR A              ; フラグセット
400E  F21740            JP P,RDDONE       ; 読込み終了の場合はRDDONEへ
4011  DBFB              IN A,(0FBH)       ; 1バイトのデータ読込み
4013  77                LD (HL),A
4014  23                INC HL            ;
4015  18F4              JR RLOOP          ;
4017  DBF8     RDDONE   IN A,(0F8H)       ; ディスクステータス読込み
4019  323040            LD (4030H),A      ; 4030H番地へストア
401C  C311F8            JP MONITOR        ; モニタへジャンプ
```

図8.16 1セクタ読込みプログラム

```
                             SOURCE STATEMENT

F811           MONITOR  EQU   0F811H
               ;
                        ORG 4100H
4100  210050   WSECTOR  LD HL,5000H       ; ライトデータの番地指定
4103  3E01              LD A,1            ;}セクタ番号1の設定
4105  D3FA              OUT (0FAH),A      ;}
4107  3EAC              LD A,0ACH         ;}書込み命令の実行
4109  D3F8              OUT (0F8H),A      ;}
410B  DBFC     WLOOP    IN A,(0FCH)       ; INTRQの間WAIT
410D  B7                OR A              ; フラグセット
410E  F21741            JP P,WDONE        ; 終了したらWDONEへ
4111  7E                LD A,(HL)         ; RAMからデータの取込み
4112  D3FB              OUT (0FBH),A      ; フロッピーディスクへの書込み
4114  23                INC HL            ;
4115  18F4              JR WLOOP          ;
4117  DBF8     WDONE    IN A,(0F8H)       ; ステータスリード
4119  323041            LD (4130H),A      ; 4130番地へストア
411C  C311F8            JP MONITOR        ; モニタへジャンプ
```

図8.17 1セクタ書込みプログラム

リードするデータをストアする番地は4001H および4002H番地のデータを変えることによって変更可能であり，またセクタ番号は4004H番地の“01”を26以内の任意の数に変えることによって変更できます。

■ライトテストプログラム

リードテストプログラムと同様に，1セクタ分のフロッピーディスクへの書込みプログラムリストを**図8.17**に示します。このプログラムは“4100H”番地から入力し，正常に動作した場合，モニタの先頭へＪＵＭＰするように構成されています。

ライト作業後のステータスは“4130H”番地にストアされます。このライトテストと前項のリードテストを同時に用いて，正しく128バイト分（5000Hから507FHまで）のデータがREAD／WRITEできることを繰り返しテストしておきましょう。

8.3 第8章　フロッピーディスク制御回路の作り方 CP／Mモニタプログラムの作り方

前節までで，フロッピーディスクに対して，1セクタ分のREAD／WRITEなどの基本操作が正常に動作することが確認できました。次に数セクタ分のREAD／WRITE作業を行ないます。ここでは，次章で述べるCP／Mの移植操作で必要となるBIOS（Basic Input Output System）を含むCP／M用モニタプログラムを作成することにします。

CP／Mモニタプログラムのリストを**図8.18**に示します。このCP／Mモニタプログラムでは，表7.1に示したモニタプログラムのコマンド群に対し，新たにフロッピーディスクへの書込み操作である“W”コマンドとフロッピーディスクからの読込み操作である“R”コマンドを追加してあります。

したがって，図8.18に示すCP／Mモニタプログラムを，第7章で作成したモニタプログラムを用いて打ち込み，EPROMへ自動書込みを行なって下さい。

このCP／Mモニタプログラムは，第7章で作成した図7.19のモニタプログラム（0000H～02DAH）に，後半の部分（02DBH～048AH）を追加した構成となっていて，前半はまったく同じ構成です。基本モニタのROMを自動ROMライタにより読込み作業を行ない，後半の部分だけをキーインして，EPROMに自動書込みを行なえば，効率よくCP／Mモニタプログラムが作成できます。

■リード動作

新たに追加したリードコマンド“R”は，モニタモードで“R”とキーインすることにより，フロッピーディスクからメモリへのデータの読込みが行なえます。モニタモードで“R”をキーインすると，“ADR>”と表示され，フロッピーディスクから読み込んだデータのストア開始番地を16進4桁でキーインします。

次に，“TRK>”と表示されますので，フロッピーディスクからリードするトラック数を0～74までの10進数2桁の数字でキーインすると，すぐにフロッピーリード動作に移ります。このリード動作が正常に終了すると，ふたたびCP／Mモニタプログラムの先頭番地にもどり，新しいコマンドの要求命令“>”が出されます。

このリードプログラムは，モニタプログラム中の319H番地から34EH番地にプログラミングされており，2Kバイト（800H）単位でリード作業を行なうように固定してあります。そのため，全トラックは1～16までの16セクタ

```
030C- FB0C  21C3F8    FDWRITE: LD HL,MMONIS    ;  FDWRITEプログラム，指定番地〔HL〕から4Kバイト
030F- FB0F  CD0DFA             CALL PMSG       ;  (16セクタ)分のデータをフロッピーディスクに書き込む
0312- FB12  CDCBF8             CALL D16IN      ;  アドレス入力
0315- FB15  22C3F4             LD (DMAADD),HL  ;
0318- FB18  2142FB             LD HL,MFDWR2    ;
031B- FB1B  CD0DFA             CALL PMSG       ;
031E- FB1E  CD80FB             CALL D10IN      ;  トラック番号入力
0321- FB21  4F                 LD C,A          ;
0322- FB22  CDA9FB             CALL SETTRK     ;
0325- FB25  1E00               LD E,0          ;
0327- FB27  018000             LD BC,128       ;
032A- FB2A  1C        FDWRT1:  INC E           ;
032B- FB2B  7B                 LD A,E          ;
032C- FB2C  32C1F4             LD (SECT),A     ;
032F- FB2F  CD90FC             CALL WRITE      ;
0332- FB32  2AC3F4             LD HL,(DMAADD)  ;  16セクタ書込みルーチン
0335- FB35  09                 ADD HL,BC       ;
0336- FB36  22C3F4             LD (DMAADD),HL  ;
0339- FB39  7B                 LD A,E          ;
033A- FB3A  FE10               CP 16           ;
033C- FB3C  C22AFB             JP NZ,FDWRT1    ;
033F- FB3F  C31CF8             JP MONIT1
0342- FB42  0A0D5452  MFDWR2:  DEFB LF,CR,'T','R','K',LF,CR,0
0346- FB46  4B0A0D00
------------------------------------------------------------------------------
034A- FB4A  21C3F8    FDREAD:  LD HL,MMONIS    ;  FDREADプログラム，指定番地から4Kバイト読み込む
034D- FB4D  CD0DFA             CALL PMSG       ;  アドレス入力
0350- FB50  CDCBF8             CALL D16IN      ;
0353- FB53  22C3F4             LD (DMAADD),HL  ;
0356- FB56  2142FB             LD HL,MFDWR2    ;
0359- FB59  CD0DFA             CALL PMSG       ;  トラック番号入力
035C- FB5C  CD80FB             CALL D10IN      ;
035F- FB5F  4F                 LD C,A          ;
0360- FB60  CDA9FB             CALL SETTRK     ;
0363- FB63  1E00               LD E,0          ;
0365- FB65  018000             LD BC,128       ;
0368- FB68  1C        FDRED1:  INC E           ;
0369- FB69  7B                 LD A,E          ;
036A- FB6A  32C1F4             LD (SECT),A     ;
036D- FB6D  CD12FC             CALL READ       ;
0370- FB70  2AC3F4             LD HL,(DMAADD)  ;  16セクタ読込みルーチン
0373- FB73  09                 ADD HL,BC       ;
0374- FB74  22C3F4             LD (DMAADD),HL  ;
0377- FB77  7B                 LD A,E          ;
0378- FB78  FE10               CP 16           ;
037A- FB7A  C268FB             JP NZ,FDRED1    ;
037D- FB7D  C31CF8             JP MONIT1
------------------------------------------------------------------------------
0380- FB80  C5        D10IN:   PUSH BC         ;  サブルーチンD10IN，10進数キーボード入力
0381- FB81  D5                 PUSH DE         ;
0382- FB82  1600               LD D,0          ;
0384- FB84  CDBCF9    D10IN1:  CALL CONIN      ;
0387- FB87  FE0D               CP CR           ;
0389- FB89  2813               JR Z,D10IN2     ;
038B- FB8B  CD23FA             CALL CONOT      ;
038E- FB8E  E60F               AND 00001111B   ;
0390- FB90  47                 LD B,A          ;
0391- FB91  7A                 LD A,D          ;
0392- FB92  CB27               SLA A           ;  (A)*2
0394- FB94  4F                 LD C,A          ;
0395- FB95  CB27               SLA A           ;
0397- FB97  CB27               SLA A           ;  (A)*8
0399- FB99  81                 ADD A,C         ;  (A)*8+(A)*2=(A)*10
039A- FB9A  80                 ADD A,B         ;
039B- FB9B  57                 LD D,A          ;
039C- FB9C  18E6               JR D10IN1
039E- FB9E  7A        D10IN2:  LD A,D          ;
039F- FB9F  D1                 POP DE
03A0- FBA0  C1                 POP BC
03A1- FBA1  C9                 RET
------------------------------------------------------------------------------
03A2- FBA2  0E00      HOME:    LD C,0          ;  以下I/O JUMP CP/Mブートプログラム
03A4- FBA4  3EFF               LD A,0FFH       ;  シーク トラック0
```

図 8.18　CP／Mモニタプログラムリスト(1)

```
03A6-FBA6  32D0F4            LD (HLSF),A         ;
03A9-FBA9  79       SETTRK:  LD A,C              ; GET NEW TRACK
03AA-FBAA  32C0F4            LD (TRK),A          ;
03AD-FBAD  C5       SEEK:    PUSH BC             ; シーク操作
03AE-FBAE  47                LD B,A              ;
03AF-FBAF  3E0A              LD A,RTCNT          ;
03B1-FBB1  32FEF4   SRETRY:  LD (SERCNT),A       ;
03B4-FBB4  DBF9              IN A,(TRACK)        ;
03B6-FBB6  D5                PUSH DE             ;
03B7-FBB7  1601              LD D,1              ; } 2msec 待ち時間
03B9-FBB9  CDFFFA            CALL TIME           ;
03BC-FBBC  D1                POP DE              ;
03BD-FBBD  B8                CP B                ;
03BE-FBBE  2002              JR NZ,NOTHR         ;
03C0-FBC0  C1       THERE:   POP BC              ;
03C1-FBC1  C9                RET                 ;
03C2-FBC2  D5       NOTHR:   PUSH DE             ;
03C3-FBC3  1601              LD D,1              ;
03C5-FBC5  CDFFFA            CALL TIME           ;
03C8-FBC8  D1                POP DE              ;
03C9-FBC9  78                LD A,B              ;
03CA-FBCA  D3FB              OUT (DDATA),A
03CC-FBCC  DBF8     BUSY:    IN A,(DSTAT)        ;
03CE-FBCE  CB47              BIT 0,A             ;
03D0-FBD0  20FA              JR NZ,BUSY          ;
03D2-FBD2  3E14              LD A,14H            ; 14H=SEEK COMMAND
03D4-FBD4  D3F8              OUT (DCOM),A        ;
03D6-FBD6  DB7E     BUSSYZ:  IN A,(DREADY)       ; C(1)ポートによるBUSSY SENCE
03D8-FBD8  CB4F              BIT 1,A             ;
03DA-FBDA  20FA              JR NZ,BUSSYZ        ;
03DC-FBDC  DBF8              IN A,(DSTAT)        ; READ STATUS
03DE-FBDE  E691              AND 10010001B       ;
03E0-FBE0  28DE              JR Z,THERE          ;
03E2-FBE2  3AFEF4            LD A,(SERCNT)       ;
03E5-FBE5  3D                DEC A               ;
03E6-FBE6  20C9              JR NZ,SRETRY        ;
03E8-FBE8  C1                POP BC              ;
03E9-FBE9  CD3AFC            CALL RECOV          ;
03EC-FBEC  18BF              JR SEEK             ;
------------------------------------------------------------------------
03EE-FBEE  79       SETSEC:  LD A,C              ; GET SECT NO
03EF-FBEF  32C1F4            LD (SECT),A         ;
03F2-FBF2  C9                RET
------------------------------------------------------------------------
03F3-FBF3  3AD0F4   HDLD:    LD A,(HLSF)         ; GET HEAD-LOAD FLAG
03F6-FBF6  B7                OR A                ;
03F7-FBF7  2811              JR Z,HDLD1          ;
03F9-FBF9  2F                CPL                 ;
03FA-FBFA  32D0F4            LD (HLSF),A         ;
03FD-FBFD  DBF9              IN A,(TRACK)        ; HEAD LOAD
03FF-FBFF  D3FB              OUT (DDATA),A       ;
0401-FC01  3E14              LD A,14H            ; SEEK COMMAND
0403-FC03  D3F8              OUT (DCOM),A        ;
0405-FC05  DBFC              IN A,(WAIT)         ;
0407-FC07  3E04     HDLDY:   LD A,4              ;
0409-FC09  C9                RET                 ;
------------------------------------------------------------------------
040A-FC0A  DBF8     HDLD1:   IN A,(DSTAT)        ;
040C-FC0C  E620              AND 00100000B       ;
040E-FC0E  28F7              JR Z,HDLDY          ;
0410-FC10  AF                XOR A               ;
0411-FC11  C9                RET                 ;
0412-FC12  3E0A     READ:    LD A,RTCNT          ; READ 1セクタ
0414-FC14  CD7BFC   PRETRY:  CALL DSKSET         ;
0417-FC17  C688              ADD A,10001000B     ; ADD READ CODE
0419-FC19  D3F8     READE:   OUT (DCOM),A        ;
041B-FC1B  DBFC     RLOOP:   IN A,(WAIT)         ;
041D-FC1D  B7                OR A                ;
041E-FC1E  F227FC            JP P,RDDONE         ; A(7)=0なら終了
0421-FC21  DBFB              IN A,(DDATA)        ; 1バイト入力
0423-FC23  77                LD (HL),A           ;
0424-FC24  23                INC HL              ;
0425-FC25  18F4              JR RLOOP            ;
```

図 8.18　CP/Mモニタプログラムリスト(2)

```
0427-FC27  DBF8     RDDONE: IN A,(DSTAT)      ; READ DISK STATUS
0429-FC29  E69D             AND 10011101B     ;
042B-FC2B  C8               RET Z             ;
042C-FC2C  CD49FC           CALL ERCHK        ;
042F-FC2F  3AFDF4           LD A,(ERCNT)      ;
0432-FC32  3D               DEC A             ;
0433-FC33  20DF             JR NZ,PRETRY      ;
0435-FC35  CD3AFC           CALL RECOV        ;
0438-FC38  18D8             JR READ           ;
043A-FC3A  21C3FC   RECOV:  LD HL,ERRMSG      ; 10回の再試行
043D-FC3D  CD0DFA           CALL PMSG         ;
0440-FC40  CDBCF9           CALL CONIN        ;
0443-FC43  FE03             CP CTRLC          ;
0445-FC45  C0               RET NZ            ;
                    ;*******************************
                    ; IF MONITOR THEN REMOVE ';'   モニタプログラムではJP MONITIとする
0446-FC46  C31CF8           JP MONIT1         ;  第9章のCP/MプログラムではJP WBOOT
                    ; IF CPM THEN REMOVE ';'       (C35AF3)に変更する
                    ;       JP WBOOT          ;
0449-FC49  57       ERCHK:  LD D,A            ;
044A-FC4A  E610             AND 00010000B     ;
044C-FC4C  2003             JR NZ,CHKSK       ;
044E-FC4E  7A               LD A,D            ;
044F-FC4F  B7               OR A              ;
0450-FC50  C9               RET
0451-FC51  3EC4     CHKSK:  LD A,0C4H         ;
0453-FC53  D3F8             OUT (DCOM),A      ;
0455-FC55  DBFC             IN A,(WAIT)       ;
0457-FC57  DBFB             IN A,(DDATA)      ;
0459-FC59  47               LD B,A            ;
045A-FC5A  DBFC     CHKS2:  IN A,(WAIT)       ;
045C-FC5C  B7               OR A              ;
045D-FC5D  CB7F             BIT 7,A           ;
045F-FC5F  2804             JR Z,CHKS3        ;
0461-FC61  DBFB             IN A,(DDATA)      ;
0463-FC63  18F5             JR CHKS2          ;
0465-FC65  DBF8     CHKS3:  IN A,(DSTAT)      ;
0467-FC67  B7               OR A              ;
0468-FC68  2805             JR Z,CHKS4        ;
046A-FC6A  CDA2FB           CALL HOME         ;
046D-FC6D  1803             JR CHKS5          ;
046F-FC6F  78       CHKS4:  LD A,B            ;
0470-FC70  D3F9             OUT (TRACK),A     ;
0472-FC72  3AC0F4   CHKS5:  LD A,(TRK)        ;
0475-FC75  CDADFB           CALL SEEK         ;
0478-FC78  7A               LD A,D            ;
0479-FC79  B7               OR A              ;
047A-FC7A  C9               RET
047B-FC7B  32FDF4   DSKSET: LD (ERCNT),A      ; DISK SET UPルーチン
047E-FC7E  3ED0             LD A,0D0H         ;
0480-FC80  D3F8             OUT (DCOM),A      ;
0482-FC82  E3               EX (SP),HL        ;
0483-FC83  E3               EX (SP),HL        ;
0484-FC84  2AC3F4           LD HL,(DMAADD)    ;
0487-FC87  3AC1F4           LD A,(SECT)       ;
048A-FC8A  D3FA             OUT (SECTP),A     ;
048C-FC8C  CDF3FB           CALL HDLD         ;
048F-FC8F  C9               RET               ;
0490-FC90  3E0A     WRITE:  LD A,RTCNT        ; 1セクタ書込みルーチン
0492-FC92  CD7BFC   WRETRY: CALL DSKSET       ;
0495-FC95  C6A8             ADD A,0A8H        ;
0497-FC97  D3F8     WRITE2: OUT (DCOM),A      ;
0499-FC99  DBFC     WLOOP:  IN A,(WAIT)       ;
049B-FC9B  B7               OR A              ;
049C-FC9C  F2A5FC           JP P,WDONE        ;
049F-FC9F  7E               LD A,(HL)         ;
04A0-FCA0  D3FB             OUT (DDATA),A
04A2-FCA2  23               INC HL            ;
04A3-FCA3  18F4             JR WLOOP          ;
04A5-FCA5  DBF8     WDONE:  IN A,(DSTAT)      ;
```

図 8.18 CP/Mモニタプログラムリスト(3)

```
04A7-FCA7  E6FD              AND 11111101B    ;
04A9-FCA9  C8                RET Z            ;
04AA-FCAA  3A49FC            LD A,(ERCHK)     ;
04AD-FCAD  3D                DEC A            ;
04AE-FCAE  CD3AFC            CALL RECOV       ;
04B1-FCB1  C390FC            JP WRITE         ;
04B4-FCB4  60       SETDMA:  LD H,B           ;
04B5-FCB5  69                LD L,C           ;
04B6-FCB6  22C3F4            LD (DMAADD),HL   ;
04B9-FCB9  C9                RET
04BA-FCBA  DB7E     CONST:  IN A,(7EH)        ;
04BC-FCBC  CB47              BIT 0,A          ;
04BE-FCBE  3EFF              LD A,0FFH        ;
04C0-FCC0  C8                RET Z
04C1-FCC1  2F                CPL              ;
04C2-FCC2  C9                RET
     FCC3  4552524F ERRMSG:  DEFB 'E','R','R','O','R','.','0',' ',0
     FCC7  522E3020
     FCCB  00
```

図 8.18　CP／Mモニタプログラムリスト(4)

(16×128＝800 H)だけを使用しています。

リード動作が正しく行なわれない場合は，ふたたび8.2節のリード動作テストをして下さい。また，フロッピーディスク装置も正常に動作するかどうか，フロッピーディスク自体もリード可能なものかを確かめて下さい。

■ライト動作

ライトコマンド“W”は，フロッピーディスクへ指定アドレスから2Kバイト分だけ書き込むプログラムです。

ライト操作用のプログラムは，CP／Mモニタの2DBH番地から318H番地までに書き込まれています。このライト動作も正常に動作すれば，CP／Mモニタの先頭へもどり，コマンド要求の“>”が表示されます。

【演習問題】

1台のフロッピーディスクで，フロピーディスクの全面コピーを行なうプログラムを作りなさい。

このプログラムは，次章のCP／M作製においてぜひ必要となるプログラムです。解法を次に示します。

【解法】

全トラック（0～76）を16トラックずつの5回に分けてコピーを行ないます（ただし，最後は残りの13トラックです）。

① はじめに“R”と表示し，ソースディスクを挿入し，スペースキーを押すと0～15トラックをメモリ内に読み込み，“W”と表示します

② コピーされるディスクを挿入後，スペースキーを押すことにより，メモリ内容を0～15トラックに書き込み，その後ふたたび“R”と表示します。

③ 同様にして，第2回目は16～31トラック，第3回目は32～47トラック，第4回目に48～63，最後に64～76までのトラックを書き込み終了します。

（解答は**付録3**にあります）

第2部第9章

CP/Mマイコンの作り方

9

第8章までで完成した手作りコンピュータに，現在たいへんに普及しているCP/Mオペレーティングシステムを移植して，CP/Mマイコンとします。ここでは，CP/Mの移植方法を述べ，活用方法について簡単に触れます。CP/Mに関する詳しい説明は多くの書籍が出版されています[9-1)9-2)]ので，そちらを参照して下さい。

9.1 CP／Mについて

コンピュータの入出力機器や実行プログラムを容易にかつ，効率良く操作する管理プログラムをOS（オペレーティングシステム）とよんでおり，大型コンピュータはむろんのこと，近年ではマイコンにも導入されています。オペレーティングシステムが入出力機器，データやプログラムの管理を行なうことにより，ユーザは入出力装置の動作を詳しく知ることなく，マイコンを操作することができるようになりました。

とくに，マイコン用の80系のOSとして主流となっているのがCP／Mで，標準オペレーティングシステムとなりつつあります。CP／Mは，ファイル管理を中心に，ファイルの作成，削除，転送などの機能が充実しており，CP／Mのもとで動作するソフトウエアには，各種アセンブラ，BASIC, FORTRAN, PL／I, PASCALなどの高級言語や各種アプリケーション用ソフトウエアパッケージが開発され，販売されています。*

■CP／Mの実装

CP／Mは標準OSとして定着していますが，その理由の一つとして，Z-80（またはインテル8080）CPUを用いていることと，少なくとも16Kバイトのメモリとディスクドライブをもつコンピュータ構成であれば，CP／Mが容易に実装できる点です。そのため，手作りコンピュータもCP／Mを実装させることにより，市販されているマイコン並みのソフトウエアを装備したコンピュータになります。

これは，CP／Mが**図 9.1** に示すようにBIOS, BDOS, CCP, TPAの4つの部分から構成され，各機種のハードウエアに依存する部分はBIOSとして扱うことができるからです。つまり，BIOS部分のみの修正により，残りのCP／Mのソフトウエアは共通に利用できる構成になっています。

■CP／Mの概略

BIOSは，ディスクのアクセスおよび標準的な周辺機器とのインターフェースに必要な，もっとも基本的な操作を行なう部分です。これについては次節9.2で，手作りシステムへの変更の手続きを示します。

BDOSは，独立なファイルディレクトリをもつディスク装置をコントロールして，ディスク

＊　「インターフェース」などの商業誌にはソフトウエアハウスより市販されているCP／Mで動くソフトウエアライブラリが紹介されており，容易に入手可能となっています。

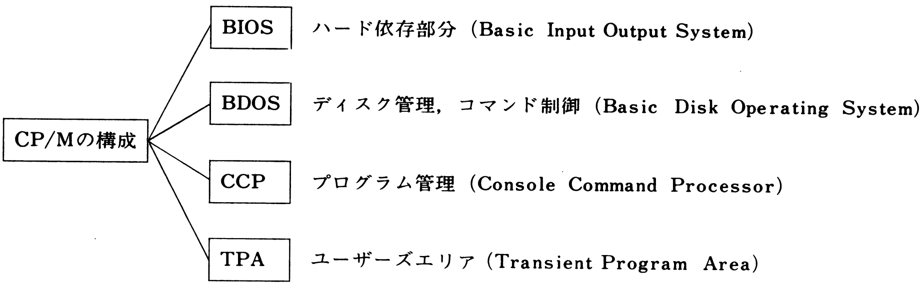


図9.1 CP／Mの構成

管理を行ないます。また，ファイルの動的割付けをするディスク割付け法を備えています。このBDOSでは，以下のような基本的操作を行ない，その内容を示すプログラムアクセス可能なエントリーポイントをもっています。

SEARCH：指定された名前により，ある特定のディスクファイルを探す
OPEN：ファイルを開く
CLOSE：ファイルを閉じる
RENAME：ファイル名の変更
READ：特定ファイルからレコードを読み出す
WRITE：特定のファイルにレコードを書き込む
SELECT：ディスクドライブの選択

CCPは，コンソールからコマンドを読み取り，ファイルディレクトリのリスティング，またファイルの内容のプリント，アセンブラ，エディタ，デバッガのようなプログラム動作のコントロールなど，コマンドの処理を行ないます。

TPAは，CCPのコマンドにより，ディスクからロードされたプログラムを保持します。例えば，プログラムの編集中，TPAはCP／Mテキスト，マシンコードとデータエリアを確保しています。

9.2 CP／Mの実装手順

第9章　CP／Mマイコンの作り方

ここでは，ターベル社製*の標準CP／Mバージョン2.0，メモリサイズ24Kを手作りシステムへ実装する手順について述べます。本書では，移植手順の例として，メモリサイズ24Kを利用しますが，必ずしも24Kである必要はありません。しかし他のメモリサイズの場合は，バイアスが異なりますので，多少注意を要します。

■改造の内容

手作りコンピュータへのCP／Mの移植は，大きく分けて次の3段階の作業から構成されています。もっとも大切な部分は，第1段階のBIOS改造部分です。

① BIOSの改造
(基本入出力部分の変更)
② コールドスタートローダの作製
③ 60K CP／Mの作製

第1段階のBIOSの改造は，前章までで動作確認をしたディスプレイ出力(CONOT)，キーボード入力(CONINおよびCONST)，フロッピーディスクリード(READ)とライト(WRITE)などの基本入出力部分を，製作した手作りコンピュータ仕様とする作業です。この作業が完了すれば，もうCP／Mの移植はほぼ完成したことになります。

第2段階のコールドスタートローダの作製は，CP／Mを利用しやすくするために，

---

* 136ページのターベル版CP／Mとディジタル・リサーチ版CP／Mを参照。

EPROMにCP／M起動用のプログラムを書き込む作業を行ないます。この作業はまた，次の60 Kメモリへの拡張の前準備ともなります。

第3段階の60 K CP／Mの作製は，CP／Mを60 Kメモリサイズに拡張する作業です。手作りコンピュータが64 KのRAMメモリ容量を有し，そのうち2 KをVRAM，また2 Kを基本プログラム領域としています。したがって，残りの60 KRAM領域をCP／Mに割り当てます。

9.2.1 BIOSの改造

CP／M移植の第1段階であるBIOSの改造手順について述べます。図9.1に示したCP／Mの4つの部分中で，機器に依存するBIOSを，手作りシステム用の新しいBIOSに変更します。

変更する作業は，①～⑩の全部で10個所あり，そのうちキーボードおよびディスプレイに関するものは次の3個所です。

① キーボード文字入力（サブルーチン名CONIN）
② キーボード入力要求チェック（サブルーチン名CONST）
③ ディスプレイ出力（サブルーチン名CONOT）

また，フロッピーディスクに関して次の5個所，

④ トラック設定（サブルーチン名SETTRK）
⑤ セクタ設定（サブルーチン名SETSEC）
⑥ アドレス設定（サブルーチン名SETDMA）
⑦ 読込み作業（サブルーチン名READ）
⑧ 書込み作業（サブルーチン名WRITE）

その他，将来のプリンタ接続を考慮して，

⑨ プリンタ文字出力（サブルーチン名LIST）

そして，後の9.2.2項で説明するウォームスタート部分のアドレス設定変更が1個所

⑩ ウォームブート内のアドレス（WBOOTのDMAADD）

の合計10個所の変更を行ないます。

この変更手続きは簡単で，**図9.2**に示すように，古いBIOSの入出力処理プログラムの部分を，手作りコンピュータ用のCP／Mモニタ作業プログラム（EPROMに書き込んである）へのCALLに変更するという手続きです*。

変更後の“CALL　0F9BCH”のF9BCH番地は，第8章で作製したCP／MモニタではキーボードGo入力処理プログラムの“CONIN”が配置されている番地を示しています。

すなわち，BIOSでのキーボード入力の実際の処理は，F9BC番地に配置した自作のプログラムで行なうように改造することです。このプログラムは数行で終わりますので，古いBIOSの残りのプログラムは，不必要ないわゆるゴミ

* ここではCALL命令を使いましたが，JUMP命令でも可能です。

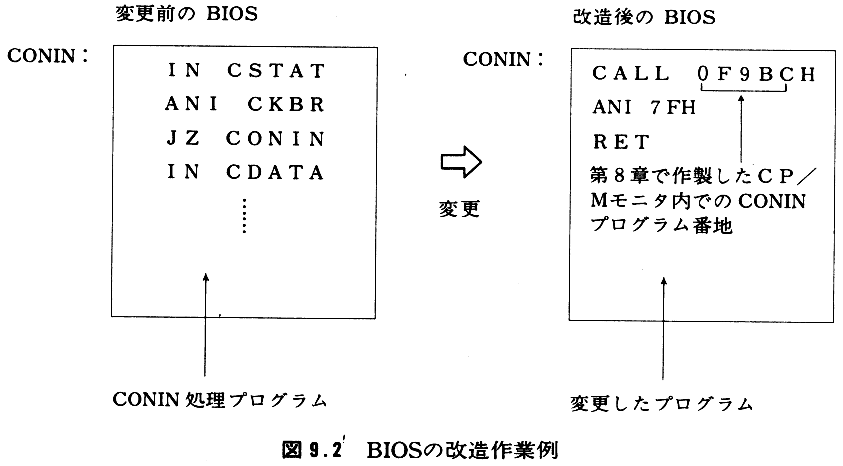


図9.2 BIOSの改造作業例

トラック番号	セクタ番号	メモリアドレス	CP／Mモジュール名
0	1	4380H (ブートアドレス)	コールドスタートローダ
0 ⋮ 0	2 ∫ 17	4400H＋b ∫ 4BFFH＋b	CCP
0 ⋮ 0 1 ⋮ 1	18 26 27 19	4C00H＋b ∫ 5000H＋b 5080H＋b ∫ 59FFH＋b	BDOS
1 ⋮ 1	20 26	5A00H＋b ∫ 5D00H＋b	BIOS

図9.3　CP／Mディスケットの最初の2トラックのセクタ内容

と称する部分となります。

■GETSYSの作製

改造に先立ち，その前準備として，BIOSをメモリ内に配置するためのプログラムを作らなければなりません。このプログラム名は，GETSYSとよび，100H番地から配置するようにすすめられています。

GETSYSプログラムの具体的な作業は，CP／Mディスケットの0と1トラックの全セクタをディスクからメモリにロードすることです。実際にはBIOSだけでなく，**図9.3**に示すように，CP／Mオペレーティングシステム全体(CCP，BDOSおよびBIOS)をメモリにロードさせます。手作りコンピュータ用に作製したGETSYSプログラムリストの例を**図9.4**に示します。

このGETSYSプログラムで大切なことは，CP／Mオペレーティングシステムをどこにロードするかのメモリアドレスの設定の仕方で，**64K内のどこに配置するべきかは入手したCP／Mのメモリサイズにより決定**しなければなりません。

一般に，メモリサイズ20Kに対して他のメモリサイズは，バイアスを考慮したものと考えられ，そのバイアス値は，

BIAS＝(メモリサイズ－20)＊1024……(1)

と計算されます。例えば24Kの場合は，バイアス値が1000Hとなります。そして，GETSYSのディスケットからのデータのメモリアドレス配置開始番地ADRSは，

ADRS＝3400H＋BIAS－80H…………(2)

で計算します。

例えば，24KメモリサイズのCP／Mディスケットを入手した場合は，バイアス値が(1)式より1000Hとなりますので，ADRS＝4380Hと計算されます。各メモリサイズとADRSの値を**表9.1**(124ページ)に示します。

この表にない場合には，入手したCP／Mのメ

```
LOC.  OBJ                   SOURCE STATEMENT

1000          BIAS     EQU   1000H          ;(MSIZE-20)*1024
4400          CCP      EQU   3400H+BIAS      入手したCP/Mのメモリ
F811          MONITOR  EQU   0F811H          サイズに従ってバイアス
00F8          DISK     EQU   0F8H           ;を変える
00F8          DCOM     EQU   DISK           ;
00F8          DSTAT    EQU   DISK           ;
00FA          SECTP    EQU   DISK+2         ;
00FB          DDATA    EQU   DISK+3
00FC          WAIT     EQU   DISK+4
008C          READ     EQU   8CH            ;
0014          SEEK     EQU   14H            ;
007E          BUSSY    EQU   7EH            ;
0800          SPTOP    EQU 800H
              ;B-REG.   TRACK COUNTER(0-76)
              ;C-REG.   SECTOR COUNTER(1-26)
              ;HL-REG.  TOP OF LOAD ADDRESS
                       ORG 100H
0100  310008  GETSYS   LD SP,SPTOP
0103  218043           LD HL,CCP-80H    ;
0106  0600             LD B,0           ;トラック番号0設定
0108  0E01    RDTRAK   LD C,1           ;セクタ番号1設定
010A  CD1C01  RDSECT   CALL READSEC     ;1セクタ読込み
010D  0C               INC C
010E  79               LD A,C           ;26セクタ分読み込む
010F  FE1B             CP 27            ;
0111  38F7             JR C,RDSECT      ;
0113  04               INC B            ;トラック番号を1にする
0114  78               LD A,B
0115  FE02             CP 2             ;
0117  38EF             JR C,RDTRAK
0119  C311F8           JP MONITOR        モニタへジャンプ
011C  78      READSEC  LD A,B
011D  D3FB             OUT (DDATA),A
011F  DBF8    BUSY     IN A,(DSTAT)     ;
0121  CB47             BIT 0,A ;
0123  20FA             JR NZ,BUSY       ;
0125  3E14             LD A,SEEK        ;
0127  D3F8             OUT (DCOM),A     ;
0129  DB7E    BUSSYZ   IN A,(BUSSY)     ;
012B  CB4F             BIT 1,A          ;
012D  20FA             JR NZ,BUSSYZ
012F  79               LD A,C
0130  D3FA             OUT (SECTP),A    ;
0132  3E8C             LD A,READ        ;
0134  D3F8    READE    OUT (DCOM),A     ;
0136  DBFC    RLOOP    IN A,(WAIT)      ;
0138  B7               OR  A            ;
0139  F24201           JP P,RDDONE      ;
013C  DBFB             IN A,(DDATA)     ;
013E  77               LD (HL),A        ;
013F  23               INC HL           ;
0140  18F4             JR RLOOP
0142  C9      RDDONE   RET
```

（011C～0142：サブルーチンセクタリード）

図 9.4 GETSYSプログラムの作製例（メモリサイズ 24 K用）

モリサイズを確認し，(2)式に従ってメモリ配置番地を計算します。このとき，GETSYSプログラムの 101 H，102 H番地および 104 H，105 H番地の内容を入手したCP／Mのメモリサイズに合致するように変更して下さい。

以下，本書では**すべて 24 Kメモリサイズを対象として説明**しますので，他のメモリサイズを改造する場合は，必ずそのバイアス分を考慮して下さい。

なお，図 9.4 に示したGETSYSプログラムでは，GETSYSの動作が正常に終了した場合は，ふたたびモニタプログラムにもどるようにしてあります。

GETSYSは重要なプログラムなので，正しく動作するかどうかの厳重なチェックが必要となります。方法としては，はじめは内容が消えて

もかまわないディスケットを用いてメモリに書き込ませ，正しく読み込まれたかどうかをチェックしなければなりません。

また，このGETSYSプログラムは繰り返し使用しますので，プログラムを100 H番地からキーインし，終了したらモニタの“W”モードで，あるトラックに格納し，随時読み出せるようにしておくと便利です。

■GETSYSの実行

このGETSYSプログラムの正常動作が確認できたら，次に入手したCP／Mディスケットを使って，GETSYSを動作させます。すなわち，モニタモードの“G”実行モードでGETSYSの開始番地 0100 Hをキーインします。このプログラムにより，2トラック分が読み込まれ，4380 H番地から配置されます。この作業では，必ずオリジナルフロッピーディスクは保存し，付録3を利用して作ったCP／Mのコピーフロッピーディスクに対して行なって下さい。

GETSYSプログラムが正常に動作し，かつメモリサイズが正しければ，BIOSの開始番地である5A 00 H以下が**図 9.5** に示すメモリ内容となっていることが確認できます。また**図 9.6** に示すように，ここには各入出力装置へのJUMP命令が配置されています。

ただし，プログラムリスト中5Aはメモリサイズにより異なることを注意して下さい。ターベル版以外のCP／Mでは，図中の□の番地が異なっています。もし，図 9.5 に示すような入出力のJUMP番地が書き込まれていなければ，入手したメモリサイズとGETSYSのスタート番地が合致していないことになります。再確認して下さい。

■PUTSYS

表 9.1　GETSYSプログラムにおける2トラック格納開始番地

入手したCP／Mのメモリサイズ	メモリ配置開始番地	バイアス
20 K	3380 H	0
24 K	4380 H	1000 H
28 K	5380 H	2000 H
⋮	⋮	⋮
48 K	A 380 H	7000 H
⋮	⋮	⋮
60 K	D 380 H	A 000 H
64 K	E 380 H	B 000 H

番地	内容
5A 00	C3
5A 01	9C
5A 02	5A
5A 03	C3
5A 04	F3
5A 05	5A
5A 06	C3
⋮	

5A はメモリサイズにより異なる

図 9.5　BIOSの開始番号からの内容

```
5A00  C39C5A              JMP  BOOT      ; FROM COLD START LOADER.
5A03  C3F35A   WBOOTE:    JMP  WBOOT     ; FROM WARM BOOT.
5A06  C34C5B              JMP  CONST     ; CHECK CONSOLE KB STATUS.
5A09  C3555B              JMP  CONIN     ; READ CONSOLE CHARACTER.
5A0C  C3615B              JMP  CONOT     ; WRITE CONSOLE CHARACTER.
5A0F  C30F5D              JMP  LIST      ; WRITE LISTING CHAR.

5A12  C31A5D              JMP  PUNCH     ; WRITE PUNCH CHAR.
5A15  C31B5D              JMP  READER    ; READ READER CHAR.
5A18  C3A75B              JMP  HOME      ; MOVE DISK TO TRACK ZERO.
5A1B  C36C5B              JMP  SELDSK    ; SELECT DISK DRIVE.
5A1E  C3A95B              JMP  SETTRK    ; SEEK TO TRACK IN REG A.
5A21  C3FF5B              JMP  SETSEC    ; SET SECTOR NUMBER.
5A24  C30A5C              JMP  SETDMA    ; SET DISK STARTING ADR.
5A27  C3315C              JMP  READ      ; READ SELECTED SECTOR.
5A2A  C3B55C              JMP  WRITE     ; WRITE SELECTED SECTOR.
5A2D  C3085D              JMP  PRSTAT    ; LIST STATUS CHECK.
5A30  C3045C              JMP  SECTRAN   ; SECTOR TRANSLATE ROUTI NE.
```

図 9.6　BIOSの 5 A 00 H番地からのアセンブラリスト

GETSYSとは逆に，新しく手作りコンピュータシステム用に作製した新BIOSをCP／Mディスケット上に書き込むプログラムがPUTSYSです。このPUTSYSプログラムの作製例のリストを，**図 9.7** に示します。PUTSYSプログラムは，200 H番地からスタートするように構成します。

次に，入手したメモリサイズのCP／Mに合わせてバイアスを考慮し，モニタプログラムの“M”モードを用いてこのPUTSYSプログラムをキーインします。その後，“W”モードで，フロッピーディスクの適当なトラック上に書き込んでおきます。このPUTSYSプログラムも正しく動作することを厳重にチェックして下さい。

■CBIOSの作製

いよいよ手作りコンピュータシステムに合わせてBIOSを変更します。この新しいBIOSをCBIOS（Customized BIOS）とよんでいます。

```
LOC.  OBJ                SOURCE STATEMENT
1000             BIAS    EQU  1000H       ;(MSIZE-20)*1024
4400             CCP     EQU  3400H+BIAS   入手したCP/Mのメモリ
F811             MONITOR EQU  0F811H       サイズに従ってバイアス
00F8             DISK    EQU  0F8H         を変える
00F8             DCOM    EQU  DISK
00F8             DSTAT   EQU  DISK
00FA             SECTP   EQU  DISK+2
00FB             DDATA   EQU  DISK+3
00FC             WAIT    EQU  DISK+4
00AC             WRITE   EQU  0ACH
0014             SEEK    EQU  14H
007E             BUSSY   EQU  7EH
0800             SPTOP   EQU  800H
                         ORG 100H
0100  310008     PUTSYS  LD SP,SPTOP
0103  218043             LD HL,CCP-80H
0106  0600               LD B,0           ;トラック番号0設定
0108  0E01       WRTRAK  LD C,1           ;セクタ番号1設定
010A  CD1C01     WRSECT  CALL WRITSEC     ;1セクタ書込み
010D  0C                 INC C
010E  79                 LD A,C
010F  FE1B               CP 27            ;
0111  38F7               JR C,WRSECT
0113  04                 INC B            ;トラック番号1の設定
0114  78                 LD A,B
0115  FE02               CP 2             ;
0117  38EF               JR C,WRTRAK
0119  C311F8             JP MONITOR
011C  78         WRITSEC LD A,B           ;
011D  D3FB               OUT (DDATA),A    ;
011F  DBF8       BUSY    IN A,(DSTAT)     ;
0121  CB47               BIT 0,A          ;
0123  20FA               JR NZ,BUSY       ;
0125  3E14               LD A,SEEK        ;
0127  D3F8               OUT (DCOM),A
0129  DB7E       BUSSYZ  IN A,(BUSSY)     ;} C(1)ポートを用いて
012B  CB4F               BIT 1,A          ;} BUSSYセンス
012D  20FA               JR NZ,BUSSYZ     ;
012F  79                 LD A,C
0130  D3FA               OUT (SECTP),A    ;
0132  3EAC               LD A,WRITE        ライトコマンド設定
0134  D3F8               OUT (DCOM),A
0136  DBFC       WLOOP   IN A,(WAIT)      ;
0138  B7                 OR A             ;
0139  F24201             JP P,WDONE       ;
013C  7E                 LD A,(HL)        ;
013D  D3FB               OUT (DDATA),A;
013F  23                 INC HL
0140  18F4               JR WLOOP
0142  C9         WDONE   RET
```

サブルーチンセクタライト

図 9.7 PUTSYSプログラムの作製例（メモリサイズ24 K用）

CBIOSの作り方は次の順序で行ないます。

① GETSYSプログラムを100H番地にロードしておく。

② 100番地から実行して，入手したCP／Mディスケットの2トラック分を読み込み，5A00H番地（ただし24Kの場合）以下に，図9.5に示すようなI／O JUMP番地が書き込まれていることを確認します。

③ 次に**表9.2**に示す各入出力動作に関する部分を手作り用コンピュータ用に訂正します。この方法は，表中に示してありますが，修正すべき番地にその訂正内容をキーインして下さい。

例えば"CONST"に対しては，5B4C番

表9.2　CBIOSの作り方

入出力 サブルーチン名	修正番地	訂正前の 内　容	訂正内容	アセンブラ
CONIN	5B55	DB	CD	CALL　CONIN
	5B56	00	BC	
	5B57	E6	F9	ANI　7FH
	5B58	01	E6	
	5B59	C2	7F	
	5B5A	55	C9	RET
CONOT	5B61	DB	79	LD　A，C
	5B62	00	CD	CALL　CONOT
	5B63	E6	F2	
	5B64	80	F9	
	5B65	C2	C9	RET
CONST	5B4C	DB	CD	CALL　CONST
	5B4D	00	BA	
	5B4E	E6	FC	
	5B4F	01	C9	RET
SET　TRK	5BA9	E5	CD	CALL SETTRK
	5BAA	2A	F9	
	5BAB	2C	FB	
	5BAC	2D	C9	RET
SET　DMA	5C0A	60	CD	CALL SETDMA
	5C0B	69	B4	
	5C0C	22	FC	
	5C0D	1E	C9	RET
READ	5C31	3E	CD	CALL　READ
	5C32	0A	12	
	5C33	CD	FC	
	5C34	A0	C9	RET
WRITE	5CB5	3E	CD	CALL　WRITE
	5CB6	0A	90	
	5CB7	CD	FC	
	5CB8	A0	C9	RET
ウォームスタートローダ	5B19	1E	C3	
	5B20	5D	F4	

地以下の内容を，作製したCP／Mモニタの“CONST”のスタート番地（FCBA）をCALLするようにCDBAFC（CALL CONST），C 9（RET）に変更します。以下同様に，表 9.2 に示す 7 つのサブルーチンへのCALL動作に変更します。

④ 最後に，WBOOTのDMAADDの番地を変更します。**図 9.8** に示すように，ウォームブートプログラム内の 5 B 18 番地～ 5 B 20 番地に配置されている命令“SHLD DMAADD”におけるDMAADDの実際の番地を，第 8 章で作製したCP／Mモニタの DMAADDの番地である“F 4 C 3 H”に変更させます。

⑤ 以上の操作により，新しいBIOSが完成したことになりますので，PUTSYSを用いて新しいフロッピーディスクに書き込んで下さい。これが，もっとも基本となる自分用のCP／Mとなります。以下，これらの変更を行なったCP／MディスケットのことをパッチずみのCP／Mとよぶことにします。

⑥ 次に，不要なフロッピーディスクをディスクドライブに挿入し，コールドブートの開始番地である 5 A 00 番地（24 K以外はバイアスを考慮のこと）から実行して下さい。コールドスタートルーチンは，ゼロページのメモリをイニシャライズした後，CCP（4400 H番地）へジャンプします。さらに，CCPはBDOSをCALLし，BDOSでは，CBIOSをCALLします。そして，③で作製したCBIOSが正しく動作すれば，CP／Mは“A>”を表示します。なお，この“A>”が表示されれば，後は，問題なくCP／Mが動きます。もし“A>”が表示されなければ，ふたたび①にもどり，厳重にチェックして下さい。

■動作テスト

変更したCP／Mシステムをロードし，

DIR ⓡ（ⓡはキャリッジリターンキー）

と入力すると，CP／Mはディスケットに書き込まれているファイルの一覧を表示します。また，

SAVE 1 X. COM ⓡ

と入力すると，CP／Mはディスクを何回かアクセスした後，“A>”と応答してきます。それから，ふたたび“DIR ⓡ”を押すと，新しくファイル“A：X COM”が作成されている様子がわかります。

さらに，

ERA X. COM ⓡ

と入力すると，X. COMのファイルは削除されることがわかります。この操作が確実に実行できれば，ほぼCP／Mは完全に動いているといえます。

■コールドスタートローダの構成

BIOSの変更が終了し，CP／Mの動作が確認されたら，次にコールドスタートローダを作ります。コールドスタートローダの動作は，図 9.4 に示したGETSYSとほぼ同様な動作をして，

変更前の BIOS

番　地	内　容
5 B 18	22
5 B 19	1 E
5 B 20	5 D

⇨ 変更

変更後の BIOS

番　地	内　容
5 B 18	22
5 B 19	C 3
5 B 20	F 4

5 B 18～20番地のアセンブラ

（変更前）
RBLK 1：SHLD DMAADD

（変更後）
RBLK 1：SHLD 0F 4 C 3 H

図 9.8 ウォームブートの変更

図 9.9 に示すように第0トラックの第1セクタを除いた第2セクタから第1トラックの第26セクタまでをメモリの4400 H（CCPの開始番地）以後にロードする作業を行ないます。このコールドスタートローダのプログラム作製例のリストを**図 9.10** に示します。

コールドスタートローダの作製手順は，次のように行ないます。

① GETSYSにより，パッチずみのCP／Mシステムを4380 Hからロードします。

```
LOC.  OBJ                    SOURCE STATEMENT
                             ;COLD START LOADER          コメントおよび
                             ;                            アセンブラ文
1000               BIAS      EQU   1000H        ;
4400               CCP       EQU   3400H+BIAS   ;
5A00               BIOS      EQU   CCP+1600H    ;
5A00               BOOT      EQU   BIOS         ;
                   ;
00F8               DISK      EQU   0F8H         ;
00F8               DCOM      EQU   DISK         ;
00F8               DSTAT     EQU   DISK         ;
00FA               SECTP     EQU   DISK+2       ;
00FB               DDATA     EQU   DISK+3       ;
00FC               WAIT      EQU   DISK+4       ;
007E               BSTAT     EQU   7EH          ;
                   ;
0000  318043                 LD SP,CCP-80H      ;
0003  210044                 LD HL,CCP          ;
0006  0600                   LD B,0             ; トラック番号0セット
0008  0E02                   LD C,2             ;
000A  C30F00                 JP RDSECT
000D               RDTRAK
000D  0E01                   LD C,1             ; セクタ番号1セット
000F               RDSECT
000F  CD2100                 CALL READSEC       ;
0012  0C                     INC C              ;
0013  79                     LD A,C             ;
0014  FE1B                   CP 27              ; セクタ番号26まで繰り返す
0016  38F7                   JR C,RDSECT        ;
0018  04                     INC B              ;
0019  78                     LD A,B             ;
001A  FE02                   CP 2               ; トラック2まで繰り返す
001C  38EF                   JR C,RDTRAK        ;
001E  C3005A                 JP BOOT            ;
                   ;                             以下サブルーチンREADSEC 1セクタ分読み出す
                   ;
0021               READSEC
0021  78                     LD A,B             ;
0022  D3FB                   OUT (DDATA),A      ;
0024  DBF8         BUSY      IN A,(DSTAT)       ;
0026  CB47                   BIT 0,A            ;
0028  20FA                   JR NZ,BUSY         ;
002A  3E14                   LD A,14H           ; シークコマンド
002C  D3F8                   OUT (DCOM),A       ;
002E  DB7E         BUSY1     IN A,(BSTAT)       ;）
0030  CB4F                   BIT 1,A            ;｝BUSSYセンス
0032  20FA                   JR NZ,BUSY1        ;）
0034  79                     LD A,C             ;
0035  D3FA                   OUT (SECTP),A      ;
0037  3E8C                   LD A,8CH           ;
0039  D3F8                   OUT (DCOM),A       ;
003B  DBFC         RLOOP     IN A,(WAIT)        ;
003D  B7                     OR A               ;
003E  F24700                 JP P,RDONE         ;
0041  DBFB                   IN A,(DDATA)       ;
0043  77                     LD (HL),A          ;
0044  23                     INC HL             ;
0045  18F4                   JR RLOOP           ;
0047  C9           RDONE     RET                ;
```

図 9.10　コールドスタートローダのプログラム

② 4380 H番地から，図 9.10 のプログラムリストに従って，コールドスタートローダのプログラムをメモリ内に書き込みます。

③ PUTSYSを使って，コールドスタートローダを追加したCP／Mシステム（4380 H以降）をフロッピーディスクのCP／Mシステムにふたたび書き込んで下さい。

以上の操作により，図 9.3 に示すように，0 トラックの 1 セクタにコールドスタートローダが書き込まれたことになります。コールドスタートローダでは，プログラムリストから明らかなように，最後にBIOSの開始番地である 5 A 00 Hへ飛び，CP／Mの動作が開始する構成となっています。

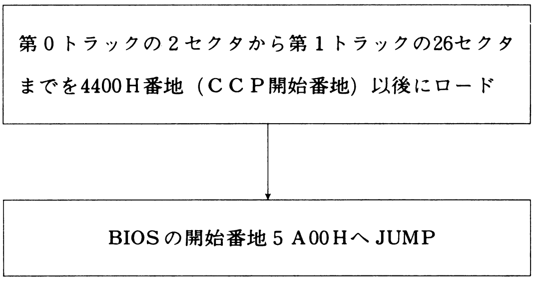


図 9.9 コールドスタートローダの動作

9.2.2 ブートストラップローダの作製

第 0 トラックの第 1 セクタにコールドスタートローダが書き込まれたCP／Mフロッピーディスクが完成したら，次に，このコールドスタートローダをメモリにロードするCP／M起動用のブートストラップローダを作ります。ブートストラップローダは，コールドスタートローダが書き込まれている第 0 トラックの第 1 セクタのみをメモリの 0 番地からロードし，ロード後にコールドスタートローダの開始番地である 0 番地にJUMPするようにプログラミングします。

手作りコンピュータシステムでは，このブートストラップローダのプログラムを，EPROM内に書き込むことにします。ブートストラップローダの作製例のプログラムリストを**図 9.11** に示します。このリストは，第 8 章で述べたCP／Mモニタの最終番地の次の番地（FCEA）から書き込みます*。

■ブートストラップの操作

EPROM 2716 に図 9.11 に示すブートストラップローダプログラムを追加書き込みし，それが終わったら，ブートプログラムの動作確認を行ないます。まずはじめに，電源を入れ，Resetスイッチを押します。

次に，ディスプレイ上に“>”が表示されたら

* ただし，FCC 3 からFCE 9 番地までは次に述べるプリンタ出力プログラムであるLISTが挿入されます。

```
04EA-  FCEA  210000   BOOT:   LD HL,0          ;メモリロード番地0設定
04ED-  FCED  010100           LD BC,1          ;セクタ番号1…Cレジスタ,トラック番号0…Bレジスタ
04F0-  FCF0  78               LD A,B           ;
04F1-  FCF1  D3FB             OUT (DDATA),A    ;
04F3-  FCF3  DBF8     BBUSY:  IN A,(DSTAT)     ;BUSSYセンス
04F5-  FCF5  CB47             BIT 0,A          ;
04F7-  FCF7  20FA             JR NZ,BBUSY      ;
04F9-  FCF9  3E14             LD A,14H         ;READコマンド
04FB-  FCFB  D3F8             OUT (DCOM),A     ;
04FD-  FCFD  DB7E     BBSSYZ: IN A,(DREADY)    ;
04FF-  FCFF  CB4F             BIT 1,A          ;
0501-  FD01  20FA             JR NZ,BBSSYZ     ;
0503-  FD03  79               LD A,C           ;
0504-  FD04  D3FA             OUT (SECTP),A    ;
0506-  FD06  3E8C             LD A,8CH         ;
0508-  FD08  D3F8     BREADE: OUT (DCOM),A     ;)
050A-  FD0A  DBFC     BRLOOP: IN A,(WAIT)      ;|
050C-  FD0C  B7               OR A             ;|
050D-  FD0D  F216FD           JP P,BROONE      ;} 0トラックの第1セクタREAD
0510-  FD10  DBFB             IN A,(DDATA)     ;|
0512-  FD12  77               LD (HL),A        ;|
0513-  FD13  23               INC HL           ;)
0514-  FD14  18F4             JR BRLOOP        ;
0516-  FD16  C30000   BROONE: JP 0              コールドスタートローダへJUMP
```

図 9.11 ブートストラップローダのプログラム

“G”を入力し，さらに開始番地であるFCEAを入力します。この入力により，コールドスタートローダがロードされ，次にBIOSが動作し，すべてが正常に動作すると，CP／Mシステムのコマンド要求である“A>”が表示されるはずです。

ここまでの作業が完全に動けば，もうほぼCP／Mシステムができたことになります。ここでふたたび，DIR ⓡとERAⓡを用いてCP／M動作テストを行ない，確実に動作することを確めて下さい。

■自動起動

ブートストラップの操作が正常に動作することが確認できたら，第8章で作製したCP／Mモニタはディスクやコンソール入出力のプログラムを除いて不必要となります。そこで，電源を投入したら，“G”命令を用いることなく，ブートストラップローダが動作するようにEPROMを書き換えます。

これは図7.19に示したモニタリスト中の1EH～20H番地を**図9.12**のように無条件にBOOTへジャンプする命令に変更すればよいことになります。この変更により，ディスプレイのイニシャライズの後，すぐにブートストラップローダに飛び，さらにコールドスタートローダがロードされ，CP／Mが起動し，“A>”が表示されることになります。

■ウォームスタート

以上述べたCP／Mの自動起動操作が正常に動作したら，Control　C(Controlキーを押しつつCを押す）を入力して下さい。この入力により，システムはウォームブートのプログラムを実行し，CP／Mシステムを再ロードした後“A>”をディスプレイ表示します。

この時点で，自分用の正しいCP／Mがフロッピーディスク上に得られたと考えられます。ウォームブートのプログラムでは，BIOSの部分を除いたCP／Mシステムの再読込みを行なった後，CCPの先頭番地へジャンプしています。

番　地	内　容
001E	C D
001F	2 3
0020	F A

アセンブラ
(変更前)
CALL　PMSG

⇨ 変更

番　地	内　容
001E	C 3
001F	E A
0020	F C

(変更後)
JUMP　BOOT

図9.12　CP／M自動起動への変更

9.2.3　プリンタの接続

前項9.2.1までの作業により，CP／Mマイコンが完成したことになります。ここでは，プリンタに関するBIOSの変更も行なったので，簡単にプリンタの接続方法について述べます。なお，プリンタの接続は，次に述べるCP／M 60 Kへの拡張作業においても非常に便利になりますので，ぜひ行なって下さい。

■プリンタの制御

キーボード制御用の8255 Aは，Bポートが使用されていませんので，データのプリンタ出力として用います。さらに，Cポートの入力用PC_3と出力用PC_6の2本を使用して，セントロニクス型のプリンタの制御を行ないます。

この配線図を**図9.13**に示します。PC_6は，出力用として**図9.14**に示すようにストローブ出力として用い，PC_3は入力なので，データ要求信号であるBUSYとして用います。このような配線にすることにより，セントロニクス型のプリンタなら制御できることになります。

■LISTプログラム

1文字をプリント出力する“LIST”サブプログラムのプログラムリストを**図9.15**に示します。プログラムは，Bポートにデータを出力の後，$\overline{\text{STROBE}}$信号を“Low”に設定し，さらに“High”にしてパルスを発生させます。次にBUSY信号が“High”となることを確認したら，1文字印字作業を終了させるプログラムとなっています。このプログラムは，CP／Mプログラムの最後の部分（ERRMSGの前）であるFCC 3H番地からFCE 0 H番地に書き込みます。

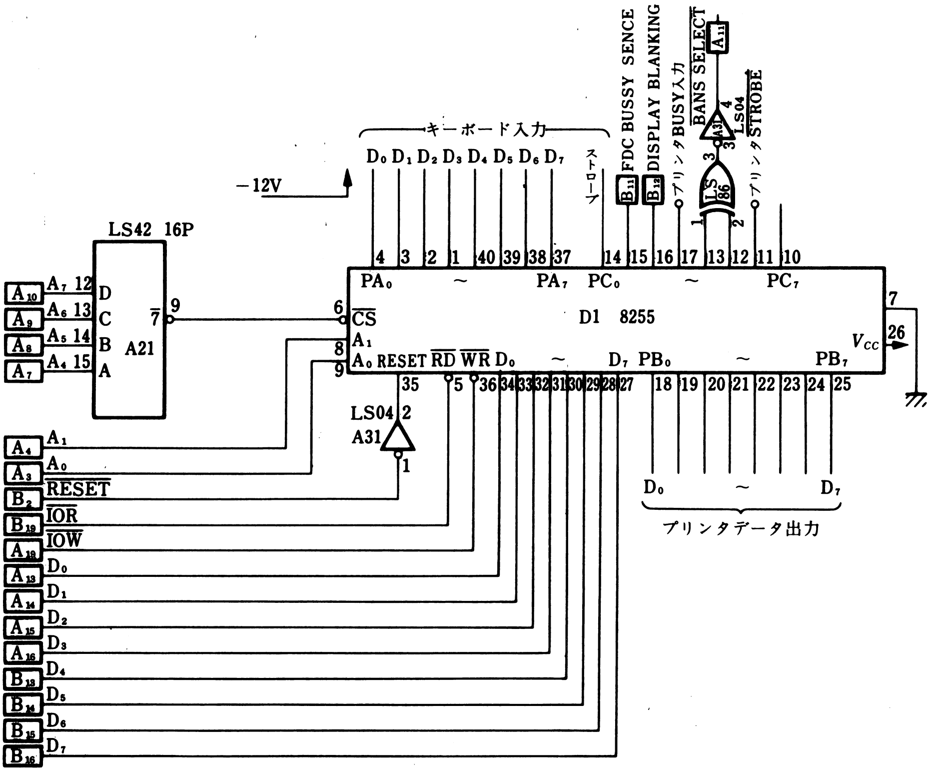


図 9.13 プリンタの接続

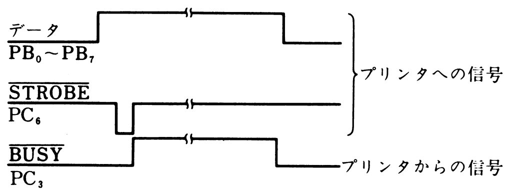


図 9.14 プリンタ制御信号のタイミング

```
                          LIST
04C3-   FCC3   F5                 PUSH AF
04C4-   FCC4   DB7E       LIST1   IN A,(7EH)      ; Cポート入力
04C6-   FCC6   CB5F               BIT 3,A         ;)
04C8-   FCC8   20FA               JR NZ,LIST1     ;) プリンタBUSSY?
04CA-   FCCA   79         LIST2   LD A,C          ;
04CB-   FCCB   D37D               OUT (7DH),A     ; Bポートデータ出力
04CD-   FCCD   3E50               LD A,50H        ;)
04CF-   FCCF   D37E               OUT (7EH),A     ;) ストローブHigh
04D1-   FCD1   3E10               LD A,10H        ;)
04D3-   FCD3   D37E               OUT (7EH),A     ;) ストローブLow
04D5-   FCD5   3E50               LD A,50H        ;)
04D7-   FCD7   D37E               OUT (7EH),A     ;) ストローブHigh
04D9-   FCD9   DB7E       LIST3   IN A,(7EH)      ;)
04DB-   FCDB   CB5F               BIT 3,A         ;) BUSSY入力待ち
04DD-   FCDD   28FA               JR Z,LIST3      ;)
04DF-   FCDF   F1                 POP AF
04E0-   FCE0   C9                 RET
```

図 9.15 プリンタ制御プログラム"LIST"（セントロンクス準拠）

9.3 CP/M60Kへの拡張

第9章 CP/Mマイコンの作り方

メモリサイズ24Kで動作するCP/Mが完成したら，最後にメモリサイズ60Kへの拡張方法について述べます。この拡張操作は，CP/Mマイコンの作り方というよりはもはや"CP/Mの使い方"に関する内容となりますので，簡単に述べることにします。

はじめてCP/Mを使う人は，まずこの拡張作業を行なう前にCP/Mシステムで動作するエディタ（ED），デバッガ（DDT：Dynamic Debugging Tool）およびアセンブラ（ASM）に慣れる必要があります。ユーザーズマニュアルを読んで，使い方に慣れてから次の拡張作業を行なって下さい。

9.3.1 CP/Mシステムの拡張作業

メモリサイズ60Kへの拡張は，**図9.16**に示すようにBIOSを変更し，CBIOSを作る作業と，第0トラックの第1セクタに挿入されるBOOTプログラムの変更およびCP/Mシステムの変更作業と，変更された3つのプログラムをつなげる作業が必要となります。各変更作業について簡単に説明します。

CCP，BDOSなどのシステムの変更，BOOTプログラムのサイズ変更，BIOSの変更に際しては，現在利用できる状態となった24K CP/Mを活用し，BIOSとBOOTは前節9.2で行なったように，機械語レベルで直接変更を行なうことはしません。まず購入してディスケットに入っているBIOS.ASMとBOOT.ASMを呼び出し，CBIOJとBOOTのソースプログラムを作製し，そのメモリサイズの部分を変更した後，アセンブラで機械語に変換します。

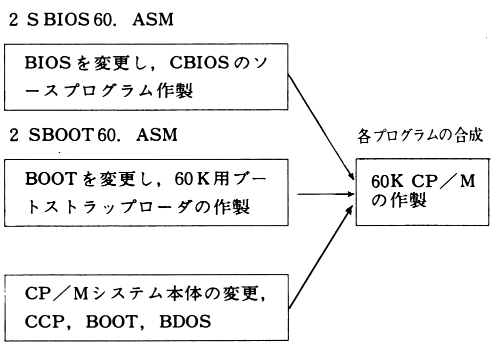


図9.16 CP/M 60Kへの拡張作業

このようにすると，任意のメモリサイズに変更するのが1個所の修正ですみます。またCBIOSの変更なども容易に行なえますので，IOBYTE制御プログラムなどの追加を行なうときにも，ソース形式で表現しておくほうが便利です。

■ 2 SBIOS 24.ASMの変更

ED（エディタ）を用いて，24K BIOSの内容を書き換えます。ここでは，9.2節で述べた各入力の変更と，メモリサイズの変更作業を行ないます。変更仕様を**図9.17**に示します。

この11個所の変更を行なったプログラムを，いったんディスクにもどし，REN（REName）により2 SBIOS 24. ASMから2 SBIOS **60**. ASMに変更します。

以上の変更が終了したら，ASMを使ってアセンブラを行ないます。次のように入力すればいいでしょう。

ASM　2 SBIOS 60 ⓡ

これで，BIOSに対しインテルの16進形式の機械コードを含む2 SBIOS 60. HEXとよばれるファイル，すなわち，

自分用のBIOSであるCBIOS

が作られたことになります。

■ 2 SBOOT 24.ASMの変更

BIOSと同様に，BOOTプログラムの変更を行ないます。EDを用いて，メモリサイズのみを24から60に変更し，2 SBOOT 60.ASMのファイルを作り，ASMを実行して2 SBOOT 24.HEXのファイルを作ります。

(1) メモリサイズの変更

```
                     60
MSIZE   EQU   24              ;MEMORY SIZE IN KBYTES.
INTRP   EQU   FALSE           ;TRUE IF INTERRUPTS ALLOWED.

STD     EQU   TRUE            ;TRUE IF STANDARD I/O.
MSIO2   EQU   FALSE           ;TRUE IF MITS 2SIO.
DELTA   EQU   FALSE           ;TRUE IF USING DELTA PRODUCTS CPU.
```

(2) WBOOTの変更

```
RDBLK:  PUSH B              ;SAVE B&C.
        MOV  C,B            ;GO TO TRACK IN B.
        CALL SETTRK
        POP  B              ;RESTORE B&C.
        JNZ  RDERR          ;IF ERROR, PRINT MESSAGE.
RBLK1:  SHLD DMAADD         ;SET STARTING ADDRESS.
        CALL SETSEC         ;READ STARTING AT SECTOR IN C.
        CALL READ
        JNZ  RDERR          ;IF ERROR, PRINT MESSAGE.
        DCR  D              ;DECREMENT SECTOR COUNT.
                 SHLD 0F4C3H
```

(3) CONSTの変更

```
                      CALL 0FCBAH
CONST:   IN    CSTAT          ;READ CONSOLE STATUS.
         ANI   CKBR           ;LOOK AT KB READY BIT.
CONST1:  MVI   A,0            ;SET A=0 FOR RETURN.
                   RET
```

(4) CONINの変更

```
CONIN:
                     CALL 0F9BCH
        IF NOT SOLOS          ;IF NOT PROC TECH SOLOS,
        IN    CSTAT           ;READ CONSOLE STATUS.
        ANI   CKBR            ;IF NOT READY,
           RET        ANI 7FH
```

(5) CONOTの変更

```
                     CALL 0FA23H
CONOT:
        IF CONUL        MOV  A,C      ;IF NULLS REQUIRED,
        MVI   A,0DH                   ;IF IT'S A CR,
        CMP   C                       ;THEN HOP OUT
        JZ    CONULL                  ;TO NULL ROUTINE.
          RET
```

(6) SETTRKの変更

```
                              CALL 0FBA9H
SETTRK:
        IF NOT DUBSID         ;IF NOT DOUBLE-SIDED,
        PUSH H                ;SAVE H&L.
        LHLD LATCH            ;GET NEW & OLD LATCH.
        MOV  A,L              ;GET NEW LATCH.
                  RET
```

(7) SETSECの変更

```
                         CALL 0FBEEH
SETSEC:  MOV   A,C            ;GET SECTOR NUMBER.
         STA   SECT           ;PUT AT SECT # ADDRESS.
         RET                  ;RETURN FROM SETSEC.
                 1行削除
```

（取り消し線の行：(1) 24、(2) SHLD DMAADD、(3) IN CSTAT・ANI CKBR、(4) IF NOT SOLOS・IN CSTAT・ANI CKBR、(5) IF CONUL・MVI A,0DH・CMP C、(6) IF NOT DUBSID・PUSH H、(7) MOV A,C・STA SECT）

図 9.17　2 SBIOS 24.ASMの変更手続き(1)

(8) SETDMAの変更

CALL 0FCB4H ／ 2行削除

```
SETDMA:  MOV  H,B          ;MOVE B&C TO H&L.
         MOV  L,C
         SHLD DMAADD       ;PUT AT DMA ADR ADDRESS.
         RET               ;RETURN FROM SETDMA.
```

(9) READの変更

CALL 0FC12H ／ RET

```
READ:    MVI  A,RTCNT      ;GET RETRY COUNT.
RRETRY:
         CALL DSKSET       ;SET UP DISK.
         ADI  88H          ;ADD CODE FOR READ SECT.
READE:   OUT  DCOM         ;SEND COMMAND TO 1771.
```

(10) WRITEの変更

CALL 0FC90H ／ RET

```
WRITE:   MVI  A,RTCNT      ;GET RETRY COUNT.
WRETRY:
         CALL DSKSET       ;SET UP DISK.
         ADI  0A8H         ;ADD CODE FOR WRITE.
```

(11) LISTの変更

CALL 0FCC3H ／ RET

```
LIST:
         IF LSTNUL         ;IF NULLS OR PAGING,
         MVI  A,0DH        ;IF IT'S A CR,
         CMP  C            ;THEN HOP OUT TO
         JZ   LINUL        ;NULL ROUTINE.
```

図 9.17 2 SBIOS 24.ASMの変更手続き(2)

■CP／Mの変更

次にCP／Mシステム全体を60Kに拡張する作業をMOVCPMプログラムを用いて行ないます。この作業により，BIOS，BOOTが変更されない前のCP／M 60Kが作られることになります。はじめに，60Kのメモリサイズ用にCP／Mのメモリイメージを TPA内に構成するために，次のコマンド

```
A>CPM 60 *
```

を与えます*。これで，CP／Mは次のように応答します。

```
CONSTRUCTING 60k CP/M vers 2.2
READY FOR "SYSGEN" OR
"SAVE 34 CPM60.COM"
```

これで，TPA内におけるCP／Mのイメージが60Kメモリサイズで得られたことになります。このメモリイメージは，0900Hから227FHの間に置かれ，**図 9.18** のように配置されています。

このメモリイメージに対し，先に作製したCBIOSとBOOTを変更すれば，60K CP／Mが完成します。この作業を行なうため，とりあえ

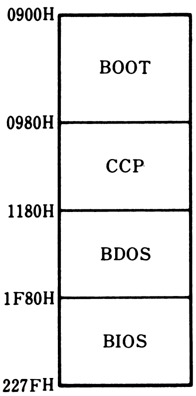


図 9.18 CP/Mシステムの
メモリイメージの配置

＊ディジタル・リサーチ版はMOVCPM 60と入力

ずこのメモリイメージを“SAVE 34 CPM 60.COM”と入力し，退避させておきます。

MOVCPMで生成されたメモリイメージは，TPAの「あき」にロードされるように負のバイアスが加えられており，これにより実際の配置番地のCP／M処理が妨げられません。

9.3.2 CP／M 60 Kシステムの作製

DDTを用いて，3つのプログラムをまとめて，オリジナルなCP／M 60 Kを作ります。DDTのもとで次の操作をします。

(1) CPM 60.COMのロード

DDT CPM 60.COMⓡ

と入力することにより，DDTをロードした後，CP／Mイメージを読み込み，DDTは次のように応答します。

NEXT PC
2300 0100

(2) BOOTのパッチ

BOOTの読込みは，“I 2 SBOOT 60.HEXⓡ R 900 Hⓡ”と入力します。ここで900 Hは負のバイアス値であり，図9.18に示すように，BOOTプログラムは900 Hから配置されます。

(3) CBIOSのパッチ

I 2 SBIOS 60. HEXとR 3580 Hを入力します。

これで60 K CP／Mがメモリイメージで完成したことになります。この際のR 3580 Hは，60Kメモリサイズに対して与えられた負のバイアス値であり，次のように計算されます。

本来，60KのメモリサイズBIOSは次の計算式，

$$\frac{(60-20)\times 1024+3400\,\text{H}+1600\,\text{H}}{\text{バイアス値}\ (=\text{A}\,000\,\text{H})}$$

$$=\text{EA}\,00\,\text{H}$$

により，EA 00 Hから配置されるようにHEXファイルは構成されています。このBIOS.HEXファイルを図9.18に示す1 F 80 Hから配置するために，そのバイアス値として3580 Hを与え，次式，

$$\text{EA}\,00\,\text{H}+3580\,\text{H}=\underline{11\,\text{F}\,80\,\text{H}}$$

残り4桁のみ有効

により，1 F 80 Hから配置されるようになります。

■SYSGENの使用

以上の操作により，60 KのCP／Mがメモリイメージ上に求まっていますので，これをCP／Mフロッピーディスク上に書きもどします。すなわち，はじめにCONTROL Cを入力し，CP／Mシステムにもどし，

SYSGEN

と入力します。これによりSYSGENが起動し，

SYSGEN VERSION 2.0
SOURCE DRIVE NAME (OR RETURN TO SKIP)

```
A>DDT CPM60.COM
DDT VERS 2.2
NEXT  PC
2300 0100
-I2SBOOT60.HEX
-R900
NEXT  PC
2300 0000
-I2SBIOS60.HEX
-R3580
NEXT  PC
2300 0000
-^C
A>SYSGEN
SYSGEN VER 2.0
SOURCE DRIVE NAME (OR RETURN TO SKIP)
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)A
DESTINATION ON A, THEN TYPE RETURN
FUNCTION COMPLETE
```

メモリサイズ60KのCP/Mプログラムの作製過程
0900H～227FHまでに配置される

ここでキャリジリターンを必ず押すこと！

作製したCP/M60KシステムのFDへの書込み作業

図9.19 60 Kシステムの作製行程

と表示されます。

ここでは，すでにシステムがメモリに入っていますので，キャリッジリターンを入力します。次に，

DESTINATION　DRIVE　NAME
(OR　RETURN　TO　REBOOT)

と表示されますので，

Aⓡ

と入力します。これにより，60Kメモリサイズの CP／Mシステムがフロッピーディスクに得られたことになります。

以上述べた60KメモリサイズのCP／Mシステムの作製に作業の行程を示したプリントリストを，**図9.19**に示します。

■CP/M 60Kのコピーの作製

CP／Mの60Kメモリサイズのシステムが作られ，正常に動作することが確認されたら，以後入手したCP／Mの下で動く各種フロッピーディスクのCP／Mシステム部分だけを60Kへ拡張する作業が必要となります。

CP／Mシステム部分のみの変更も，SYSGENを用いて次のように行なうことができます。

① はじめに，SYSGENを起動させると，

SYSGEN　VERSION　2.0
SOURCE　DRIVE　NAME　(OR
RETURN　TO　SKIP)

と表示されますので，先に作製したCP／M 60Kのディスケットを挿入し，

A ⓡ

と入力します。

② 次に，

DESTINATION　DRIVE　NAME
(OR　RETURN　TO　REBOOT)

と表示されますので，コピーしたいディスケットに交換し，

A ⓡ

と入力します。この作業により，CP／M 60Kのシステム部分のみがコピーされたことになります。

9.3.3 CP／Mの応用

以上述べた操作により，CP／M 60Kシステムが得られると，CP／Mの下で動作する各種システムが利用できることになります。その方法としては，購入したCP／Mの下で動くディスケットに対して，SYSGENを起動させて，CP／M 60Kの部分のみの変更を行なえばいいわけです。また，必要なプログラムのみの利用は，CP／Mシステム内のEDやDDTを用いても当然行なえることになります。

CP／Mの使い方を学び，有効に手作りコンピュータが利用されることを心から望みます。

《ターベル社製のCP／Mとディジタル・リサーチ社製のCP／M》

ここで作製しているマイコンでは，ターベル社製のCP／Mを実装する手続きについて，具体的に示しています。購入するCP／Mとしては，ターベル社製でもディジタル・リサーチ社製でもかまいません。

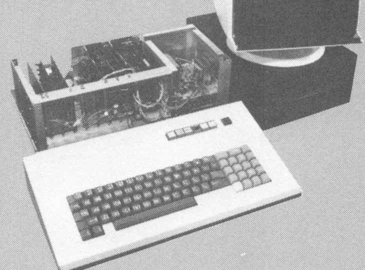
ディジタル・リサーチ社製を購入した場合も9.2.1項のBIOSの改造手順に従って，①～⑨に示す各サブルーチンの変更手続きと，⑩に示すウォームブート内のアドレスの変更を行なって下さい。この場合，ターベル版ではウォームブート内のアドレス名は“DMAADD”となっていますが，MDS版などのディジタル・リサーチ社製のCP／Mは，“DMAADR”となっていることに注意して下さい。

また，ターベル版CP／Mでは，初期サイズが24Kで供給されていますが，ディジタル・リサーチ社製のものは，20Kサイズとして供給されています。したがって，1000H (4Kバイト) のバイアス分を考慮しなければなりません。

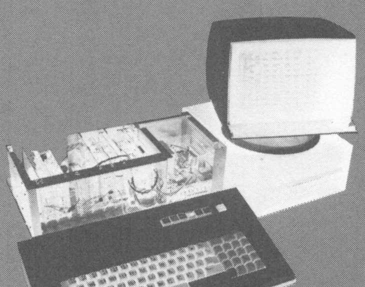
第1部
CP/Mマイコンのハード作製
第2部
CP/Mマイコンのソフト設計

第3部　付　録

第1部　CP/Mマイコンのハード作製
第2部　CP/Mマイコンのソフト設計
第3部　付　録



第1部　CP/Mマイコンのハード作製
第2部　CP/Mマイコンのソフト設計
第3部　付　録



付録 1　Z-80と8080のアセンブラ命令

(1)　8ビットのデータ移動命令

Z-80命令略号	動　作	命令コード			備　考	8080命令略号
		76	543	210		
LD r, r′	r←r′	01	r	r′	r, r′　レジスタ	MOV r, r′
LD r, n	r←n	00	r	110	000　B	MVI r, n
		←	n	→	001　C	
LD r, (HL)	r←(HL)	01	r	110	010　D	MOV r, M
LD r, (IY+d)	r←(IX+d)	11	011	101	011　E	
		01	r	110	100　H	(MOVe)
		←	d	→	101　L	
LD r, (IY+d)	r←(IY+d)	11	111	101	111　A	
		01	r	110		
		←	d	→		
LD (HL), r	(HL)←r	01	110	r		MOV M, r
LD (IX+d), r	(IX+d)←r	11	011	101		
		01	110	r		
		←	d	→		
LD (IY+d), r	(IY+d)←r	11	111	101		
		01	110	r		
		←	d	→		
LD (HL), n	(HL)←n	00	110	110		MVI M, n
		←	n	→		
LD (IX+d), n	(IX+d)←n	11	011	101		(MoVe Immediate)
		00	110	110		
		←	d	→		
		←	n	→		
LD (IY+d), n	(IY+d)←n	11	111	101		
		00	110	110		
		←	d	→		
		←	n	→		
LD A, (BC)	A←(BC)	00	001	010		LDAX B
LD A, (DE)	A←(DE)	00	011	010		LDAX D
LD A, (nn)	A←(nn)	00	111	010		LDA nn
		←	n	→		
		←	n	→		
LD (BC), A	(BC)←A	00	000	010		STAX B
LD (DE), A	(DE)←A	00	010	010		STAX D
LD (nn), A	(nn)←A	00	110	010		STA nn
		←	n	→		
		←	n	→		(STore Accumulator
LD A, I	A←1	11	101	101		direct)
		01	010	111		
LD A, R	A←R	11	101	101		
		01	011	111		
LD I, A	1←A	11	101	101		
		01	000	111		
LD R, A	R←A	11	101	101		
		01	001	111		

(2) 16ビットのデータ移動命令

Z-80命令略号	動　作	命令コード			備　考	8080命令略号
		76	543	210		
LD dd, nn	dd ← nn	00	dd0	001	dd レジスタ	LXI B, nn
		←	n	→	00 BC	LXI D, nn
		←	n	→	01 DE	LXI H, nn
LD IX, nn	IX ← nn	11	011	101	10 HL	LXI SP, nn
		00	100	001	11 SP	
		←	n	→		
		←	n	→		
LD IY, nn	IY ← nn	11	111	101		
		00	100	001		
		←	n	→		
		←	n	→		
LD HL, (nn)	H ← (nn+1)	00	101	010		LHLD nn
	L ← (nn)	←	n	→		
		←	n	→		
LD dd, (nn)	dd_H ← (nn+1)	11	101	101		
	dd_L ← (nn)	01	dd1	011		
		←	n	→		
		←	n	→		
LD IX, (nn)	IX_H ← (nn+1)	11	011	101		
	IX_L ← (nn)	00	101	010		
		←	n	→		
		←	n	→		
LD IY, (nn)	IY_H ← (nn+1)	11	111	101		
	IY_L ← (nn)	00	101	010		
		←	n	→		
		←	n	→		
LD (nn), HL	(nn+1) ← H	00	100	010		SHLD nn
	(nn) ← L	←	n	→		
		←	n	→		
LD (nn), dd	(nn+1) ← dd_H	11	101	101		
	(nn) ← dd_L	01	dd0	011		
		←	n	→		
		←	n	→		
LD (nn), IX	(nn+1) ← IX_H	11	011	101		
	(nn) ← IX_L	00	100	010		
		←	n	→		
		←	n	→		
LD (nn), IY	(nn+1) ← IY_H	11	111	101		
	(nn) ← IY_L	00	100	010		
		←	n	→		
		←	n	→		
LD SP, HL	SP ← HL	11	111	001		SPHL
LD SP, IX	SP ← IX	11	011	101		
		11	111	001		
LD SP, IY	SP ← IY	11	111	101		
		11	111	001	qq レジスタ	
PUSH qq	(SP−2) ← qq_L	11	qq0	101	00 BC	PUSH B
	(SP−1) ← qq_H				01 DE	PUSH D
PUSH IX	(SP−2) ← IX_L	11	011	101	10 HL	PUSH H
	(SP−1) ← IX_H	11	100	101	11 AF	PUSH PSW
PUSH IY	(SP−2) ← IY_L	11	111	101		
	(SP−1) ← IY_H	11	100	101		
POP qq	qq_H ← (SP+1)	11	qq0	001		POP B
	qq_L ← (SP)					POP D
POP IX	IX_H ← (SP+1)	11	011	101		POP H
	IX_L ← (SP)	11	100	001		POP PSW
POP IY	IY_H ← (SP+1)	11	111	101		
	IY_L ← (SP)	11	100	001		

(3) 交換，ブロック転送および検索命令

Z-80 命令略号	動　　作	命令コード			備　　考	8080命令略号
		76	543	210		
EX DE,HL	DE↔HL	11	101	011		XCHG
EX AF,AF′	AF↔AF′	00	001	000		
EXX	⎛BC　BC′⎞ ⎜DE ↔ DE′⎟ ⎝HL　HL′⎠	11	011	001		
EX (SP),HL	H↔(SP+1)	11	100	011		XTHL
	L↔(SP)					(eXchange H and
EX (SP),IX	IX_H↔(SP+1)	11	011	101		L With Top of
	IX_L↔(SP)	11	100	011		stack)
EX (SP),IY	IY_H↔(SP+1)	11	111	101		
	IY_L↔(SP)	11	100	011		
LDI	(DE)←(HL)	11	101	101		
	DE←DE+1	10	100	000		
	HL←HL+1					
	BC←BC−1					
LDIR	(DE)←(HL)	11	101	101		
	DE←DE+1	10	110	000		
	HL←HL+1					
	BC←BC−1					
	BC＝0まで繰り返す					
LDD	(DE)←(HL)	11	101	101		
	DE←DE−1	10	101	000		
	HL←HL−1					
	BC←BC−1					
LDDR	(DE)←(HL)	11	101	101		
	DE←DE−1	10	111	000		
	HL←HL−1					
	BC←BC−1					
	BC＝0まで繰り返す					
CPI	A−(HL)	11	101	101		
	HL←HL+1	10	100	001		
	BC←BC−1					
CPIR	A−(HL)	11	101	101		
	HL←HL+1	10	110	001		
	BC←BC−1					
	A＝(HL)または BC＝0まで繰り返す					
CPD	A−(HL)	11	101	101		
	HL←HL−1	10	101	001		
	BC←BC−1					
CPDR	A−(HL)	11	101	101		
	HL←HL−1	10	111	001		
	BC←BC−1					
	A＝(HL)または BC＝0まで繰り返す					

(4) 8ビット算術および論理演算命令

Z-80命令略号	動　作	命令コード 76	543	210	備　考	8080命令略号
ADD A, r	A←A+r	10	[000]	r	r　レジスタ	ADD N
ADD A, n	A←A+n	11	[000]	110	000　B	ADI n　N | レジスタ名
					001　C	B
		←	n	→	010　D	C
ADD A, (HL)	A←A+(HL)	10	[000]	110	011　E	ADD M　D
ADD A, (IX+d)	A←A+(IX+d)	11	011	101	100　H	E
		10	[000]	110	101　L	H
		←	d	→	111　A	L
ADD A, (IY+d)	A←A+(IY+d)	11	111	101		M
		10	[000]	110		A
		←	d	→		
ADC A, s	A←A+s+CY		[001]		sはr, n, (HL), (IX+d),	ADC N, ACI n
SUB s	A←A−s		[010]		(IY+d)のどれかを示す。	SUB N, SBI n
SBC A, s	A←A−s−CY		[011]		命令コード欄のビットは	SBB N, SBI n
AND s	A←A∧s		[100]		ADD命令の [000] と入れ換	ANA N, ANI n
OR s	A←A∨s		[110]		える。	ORA N, ORI n
XOR s	A←A⊕s		[101]			XRA N, XRI n
CP s	A−s		[111]			CMP N, CPI n
INC r	r←r+1	00	r	[100]		INR N
INC (HL)	(HL)←(HL)+1	00	110	[100]		
INC (IX+d)	(IX+d)←	11	011	101		
	(IX+d)+1	00	110	[100]		
		←	d	→		
INC (IY+d)	(IY+d)←	11	111	101		
	(IY+d)+1	00	110	[100]		
		←	d	→		
DEC d	d←d−1			[101]	dはr, (HL), (IX+d),	DCR N
					(IY+d)のどれかを示す。	
					命令コード欄のビット [101]	
					はINC命令の [100] と入れ	
					換える。	

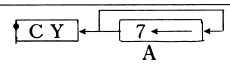
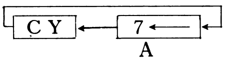
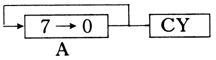
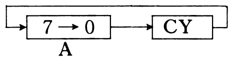
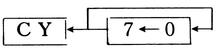
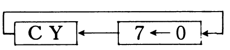
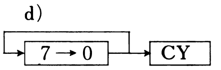
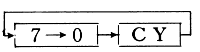
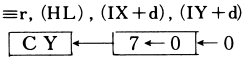
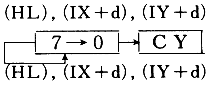
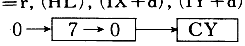
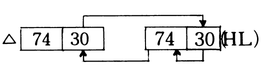
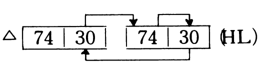
(5) 汎用算術演算およびCPU制御命令

Z-80命令略号	動　　作	命令コード			備　　考	8080命令略号
		76	543	210		
DAA	加減算命令の後に実行し，アキュムレータの内容をBCDに変換する	00	100	111	(Decimal adjust accumulator)	DAA
CPL	A←$\overline{A}$	00	101	111	(Complement accumulator) 1の補数	CMA
NEG	A←0−A	11	101	101	(Negate acc.)	
		01	000	100	2の補数	
CCF	CY←$\overline{CY}$	00	111	111	(Complement carry flag)	CMC
SCF	CY←1	00	110	111	(Set carry flag)	STC
NOP	動作せず	00	000	000		NOP
HALT	CPUを停止状態にする	01	110	110		HLT
DI	IFF←0	11	110	011		DI
EI	IFF←1	11	111	011		EI
IM 0	割込みモード0設置	11	101	101		
		01	000	110		
IM 1	割込みモード1設置	11	101	101		
		01	010	110		
IM 2	割込みモード2設置	11	101	101		
		01	011	110		

(6) 16ビット算術演算命令

Z-80命令略号	動　　作	命令コード			備　　考	8080命令略号
		76	543	210		
ADD HL, ss	HL←HL+ss	00	ss1	001	ss　レジスタ	DAD B
					00　BC	DAD D
ADC HL, ss	HL←HL+ss+CY	11	101	101	01　DE	DAD H
		01	ss1	010	10　HL	DAD SP
					11　SP	
SBC HL, ss	HL←HL−ss−CY	11	101	101		
		01	ss0	010		
ADD IX, pp	IX←IX+pp	11	011	101	pp　レジスタ	
		00	pp1	001	00　BC	
					01　DE	
					10　IX	
					11　SP	
ADD IY, rr	IY←IY+rr	11	111	101	rr　レジスタ	
		00	rr1	001	00　BC	
					01　DE	
					10　IY	
					11　SP	
INC ss	ss←ss+1	00	ss0	011		INX B
INC IX	IX←IX+1	11	011	101		INX D
		00	100	011		INX H
INC IY	IY←IY+1	11	111	101		INX SP
		00	100	011		
DEC ss	ss←ss−1	00	ss1	011		DCX B
DEC IX	IX←IX−1	11	011	101		DCX D
		00	101	011		DCX H
DEC IY	IY←IY−1	11	111	101		DCX SP
		00	101	011		

(7) 循環および桁移動命令

Z-80命令略号	動 作	命令コード 76	543	210	備 考	8080命令略号
RLCA		00	000	111	(Rotate left circular accumulator)	RLC
RLA		00	010	111	(Rotate left accumulator)	RAL
RRCA		00	001	111	(Rotate right circular accumulator)	RRC
RRA		00	011	111	(Rotate right accumulator)	RAR
RLC r		11 00	001 000	011 r	(Rotate left circular registerr)	
RLC (HL)		11 00	001 000	011 110	r レジスタ 000 B	
RLC (IX+d)	 r, (HL), (IX+d), (IY+d)	11 11 ← 00	011 001 d 000	101 011 ← 110	001 C 010 D 011 E 100 H 101 L 111 A	
RLC (IY+d)		11 11 ← 00	111 001 d 000	101 011 → 110		
RL s	 s≡r, (HL), (IX+d), (IY+d)		010		命令形式および状態はRLC rと同じ。RLC rの命令コード 000 と各命令の命令コード欄のコードと入れ換える。	
PRC s	 s≡r, (HL), (IX+d), (IY+d)		001			
RR s	 s≡r, (HL), (IX+d), (IY+d)		011			
SLA s	 s≡r, (HL), (IX+d), (IY+d)		100			
SRA s	 s≡r, (HL), (IX+d), (IY+d)		101			
SRL s	 s≡r, (HL), (IX+d), (IY+d)		111			
RLD		11 01	101 101	101 111	Aと(HL)の内容を4ビット単位で左右に回転する。Aの上位4ビットの内容は変化しない。	
RRD		11 01	101 100	101 111		

(8) ビット設定および判定命令

Z-80命令略号	動作	命令コード 76	543	210	備考		8080命令略号
BIT b, r	Z←r_b	11	001	011	r	レジスタ	
		01	b	r			
BIT b, (HL)	Z←$\overline{(HL)}_b$	11	001	011	000	B	
		01	b	110	001	C	
BIT b, (IX+d)	Z←$\overline{(IX+d)}_b$	11	011	101	010	D	
		11	001	011	011	E	
		←	d	→	100	H	
		01	b	110	101	L	
					111	A	
					b	ビット	
BIT b, (IY+d)	Z←$(IY+d)_b$	11	111	101	000	0	
		11	001	011	001	1	
		←	d	→	010	2	
		01	b	110	011	3	
					100	4	なし
					101	5	
					110	6	
					111	7	
SET b, r		11	001	011			
		[11]	b	r			
SET b, (HL)		11	001	011			
		[11]	b	110			
SET b, (IX+d)		11	011	101			
		11	001	011			
		←	d	→			
		[11]	b	110			
SET b, (IY+d)	$(IY+d)_b$←1	11	111	101			
		11	001	011			
		←	d	→	RES b, sの命令コードは		
		11	b	110	SET b, rの命令コードの[11]		
RES b, s	s_b←0	10			を[10]に入れ換えたものである。		
	s ≡ r, (HL), (IX+d), (IY+d)						

(9) 飛び越し命令

Z-80命令略号	動作	命令コード 76	543	210	備考	8080命令略号
JP nn	PC←nn	11	000	011		JMP nn
		←	n	→		
		←	n	→	cc　条件	
JP cc, nn	条件ccが成立するとPC←nnその他は飛び越さない	11	cc	010	000　NZ　ゼロでない	JNZ nn
		←	n	→	001　Z　ゼロ	JZ nn
		←	n	→	010　NC　桁上げがない	JNC nn
					011　C　桁上げがある	JC nn
					100　PO　奇パリティ	JPO nn
					101　PE　偶パリティ	JPE nn
					110　P　正	JP nn
JR e	PC←PC+e	00	011	000	111　M　負	JM nn
		←e−2		→		
JR C, e	CY＝0のとき飛び越さない	00	111	000	条件が成立しないとき	
		←e−2		→		
	CY＝1のとき PC←PC+e				条件が成立したとき	
JR NC, e	CY＝1のとき飛び越さない	00	110	000		
		←e−2		→		
	CY＝0のとき PC←PC+e					
JR Z, e	Z＝0のとき飛び越さない	00	101	000		
		←e−2		→		
	Z＝1のとき PC←PC+e					
JR NZ, e	Z＝1のとき飛び越さない	00	100	000		
		←e−2		→		
	Z＝0のとき PC←PC+e					
JP (HL)	PC←HL	11	101	001		PCHL
JP (IX)	PC←IX	11	011	101		
		11	101	001		
JP (IY)	PC←IY	11	111	101		
		11	101	001		
DJNZ e	B←B−1	00	010	000		
	B＝0のとき飛び越さない	←e−2		→		
	B≠0のとき PC←PC+e					

(10) 入力および出力命令

Z-80命令略号	動　作	命令コード			備　　考	8080命令略号
		76	543	210		
IN A, (n)	A←(n)	11	011	011	nを A_0−A_7 へ	IN n
		←	n	→	Aを A_8−A_{15}へ	
IN r, (C)	r←(C)	11	101	101	Cを A_0−A_7 へ	
	r＝110 のときフラグにのみ，影響を与える	01	r	000	Bを A_8−A_{15}へ	
INI	(HL)←(C)	11	101	101	Cを A_0−A_7 へ	
	B←B－1	10	100	010	Bを A_8−A_{15}へ	
	HL←HL＋1					
INIR	(HL)←(C)	11	101	101	Cを A_0−A_7 へ	
	B←B－1	10	110	010	Bを A_8−A_{15}へ	
	HL←HL＋1					
	B＝0になるまで繰返す					
IND	(HL)←(C)	11	101	101	Cを A_0−A_7 へ	
	B←B－1	10	101	010	Bを A_8−A_{15}へ	
	HL←HL－1					
INDR	(HL)←(C)	11	101	101	Cを A_0−A_7 へ	
	B←B－1	10	111	010	Bを A_8−A_{15}へ	
	HL←HL－1					
	B＝0になるまで繰り返す					
OUT(n),A	(n)←A	11	010	011	nを A_0−A_7 へ	OUT n
		←	n	→	Aを A_8−A_{15}へ	
OUT(C),r	(C)←r	11	101	101	Cを A_0−A_7 へ	
		01	r	001	Bを A_8−A_{15}へ	
OUTI	(C)←(HL)	11	101	101	Cを A_0−A_7 へ	
	B←B－1	10	100	011	Bを A_8−A_{15}へ	
	HL←HL＋1					
OTIR	(C)←(HL)	11	101	101	Cを A_0−A_7 へ	
	B←B－1	10	110	011	Bを A_8−A_{15}へ	
	HL←HL＋1					
	B＝0になるまで繰り返す					
OUTD	(C)←(HL)	11	101	101	Cを A_0−A_7 へ	
	B←B－1	10	101	011	Bを A_8−A_{15}へ	
	HL←HL－1					
OTDR	(C)←(HL)	11	101	101	Cを A_0−A_7 へ	
	B←B－1	10	111	011	Bを A_8−A_{15}へ	
	HL←HL－1					
	B＝0になるまで繰り返す					

(11) サブルーチン接続およびもどり命令

Z-80命令略号	動作	命令コード 76	543	210	備考	8080アセンブラ命令略号
CALL nn	(SP−1)←PC_H (SP−2)←PC_L PC←nn	11 ← ←	001 n n	101 → →		CALL nn
CALL cc, nn	条件ccが成立しないときはコール命令を無視する。条件ccが成立したときはCALL nnと同じ動作をする	11 ← ←	cc n n	100 → →		CALL cc（下表）
RET	PC_L←(SP) PC_H←(SP+1)	11	001	001		RET
RET cc	条件ccが成立しないときは次の命令を実行し，条件ccが成立したときはRETと同じ動作	11	cc	000		RET cc（下表）
RETI	割込み処理ルーチンからの戻り	11 01	101 001	101 101		
RETN	マスク不能割込み処理ルーチンからの戻り	11 01	101 000	101 101		
RST p	(SP−1)←PC_H (SP−2)←PC_L PC_H←0 PC_L←P	11	t	111		RST（下表）

cc	条件		CALL cc	RET cc
000	NZ	ゼロでない	CNZ nn	RNZ
001	Z	ゼロ	CZ nn	RZ
010	NC	桁上げがない	CNC nn	RNC
011	C	桁上げがある	CC nn	RC
100	PO	奇パリティ	CPO nn	RPO
101	PE	偶パリティ	CPE nn	RPE
110	P	正	CP nn	RP
111	M	負	CM nn	RM

t	P	
000	00H	RST 0
001	08H	RST 1
010	10H	RST 2
011	18H	RST 3
100	20H	RST 4
101	28H	RST 5
110	30H	RST 6
111	38H	RST 7

(12) アセンブラ擬似命令

ザイログ命令略号	8080命令略号	ソード略号命令	備考
DEFB	DB	無指定	1バイト定数
DEFS	DS	.BLK	領域確保
DEFW	DW	DH	2バイト定数
DEFM	DB		
DEFL	SET		ネーム割当て
COND	IFT	.IFE	）条件アセンブラ
ENDC	ENDIF	.ENDC	
GLOBAL	PUBLIC		他のプログラムでも使用可とする
EXTERNAL	EXTRN	.EXT	他のプログラムを参照するとき

付録 2　各種アセンブラ 16進↔ニーモニック 対応表

(1)　16進対応表①

16進	ザイログZ-80	8080	ソードZ-80
0 0	NOP	NOP	NOP
0 1	LD BC, nn	LXI B, nn	LXI B, nn
0 2	LD(BC), A	STAX B	STAX B
0 3	INC BC	INX B	INX B
0 4	INC B	INR B	INC B
0 5	DEC B	DCR B	DEC B
0 6	LD B, n	MVI B, n	LDI B, n
0 7	RLCA	RLC	RLC
0 8	EX AF, AF'	—	SWAPA
0 9	ADD HL, BC	DAD B	DAD B
0 A	LD A, (BC)	LDAX B	LDAX B
0 B	DEC BC	DCX B	DCX B
0 C	INC C	INR C	INC C
0 D	DEC C	DCR C	DEC C
0 E	LD C, n	MVI C, n	LDI C, n
0 F	RRCA	RRC	RRC
1 0	DJNZ e	—	DBNZ e
1 1	LD DE, nn	LXI D, nn	LXI D, nn
1 2	LD(DE), A	STAX D	STAX D
1 3	INC DE	INX D	INX D
1 4	INC D	INR D	INC D
1 5	DEC D	DCR D	DEC D
1 6	LD D, n	MVI D, n	LDI D, n
1 7	RLA	RAL	RAL
1 8	JR e	—	BR e
1 9	ADD HL, DE	DAD D	DAD D
1 A	LD A, (DE)	LDAX D	LDAX D
1 B	DEC DE	DCX D	DCX D
1 C	INC E	INR E	INC E
1 D	DEC E	DCR E	DEC E
1 E	LD E, n	MVI E, n	LDI E, n
1 F	RRA	RAR	RAR
2 0	JR NZ, e	—	BNZ e
2 1	LD HL, nn	LXI H, nn	LXI H, nn
2 2	LD(nn), HL	SHLD nn	SHLD nn
2 3	INC HL	INX H	INX H
2 4	INC H	INR H	INC H
2 5	DEC H	DCR H	DEC H
2 6	LD H, n	MVI H, n	LDI H, n
2 7	DAA	DAA	DAA
2 8	JR Z, e	—	BZZ e
2 9	ADD HL, HL	DAD H	DAD H
2 A	LD HL, (nn)	LHLD nn	LHLD nn
2 B	DEC HL	DCX H	DCX H
2 C	INC L	INR L	INC L
2 D	DEC L	DCR L	DEC L
2 E	LD L, n	MVI L, n	LDI L, n
2 F	CPL	CMA	CMA

(次ページへ続く)

(1) 16進対応表①

16進	ザイログZ-80	8080	ソードZ-80
30	JR NC, e	—	BZC e
31	LD SP, nn	LXI SP, nn	LXI SP, nn
32	LD(nn), A	STA nn	STA nn
33	INC SP	INX SP	INX SP
34	INC (HL)	INR M	INC M
35	DEC (HL)	DCR M	DEC M
36	LD(HL), n	MVI M, n	STI M, n
37	SCF	STC	STC
38	JR C, e	—	BNC e
39	ADD Hb, SP	DAD SP	DAD SP
3A	LD A, (nn)	LDA nn	LDA nn
3B	DEC SP	DCX SP	DCX SP
3C	INC A	INR A	INC A
3D	DEC A	DCR A	DEC A
3E	LD A, n	MVI A, n	LDI A, n
3F	CCF	CMC	CMC
40	LD B, B	MOV B, B	MOV B, B
41	LD B, C	MOV B, C	MOV C, B
42	LD B, D	MOV B, D	MOV D, B
43	LD B, E	MOV B, E	MOV E, B
44	LD B, H	MOV B, H	MOV H, B
45	LD B, L	MOV B, L	MOV L, B
46	LD B, (HL)	MOV B, M	LDR B, M
47	LD B, A	MOV B, A	MOV A, B
48	LD C, B	MOV C, B	MOV B, C
49	LD C, C	MOV C, C	MOV C, C
4A	LD C, D	MOV C, D	MOV D, C
4B	LD C, E	MOV C, E	MOV E, C
4C	LD C, H	MOV C, H	MOV H, C
4D	LD C, L	MOV C, L	MOV L, C
4E	LD C, (HL)	MOV C, M	LDR C, M
4F	LD C, A	MOV C, A	MOV A, C
50	LD D, B	MOV D, B	MOV B, D
51	LD D, C	MOV D, C	MOV C, D
52	LD D, D	MOV D, D	MOV D, D
53	LD D, E	MOV D, E	MOV E, D
54	LD D, H	MOV D, H	MOV H, D
55	LD D, L	MOV D, L	MOV L, D
56	LD D, (HL)	MOV D, M	LDR D, M
57	LD D, A	MOV D, A	MOV A, D
58	LD E, B	MOV E, B	MOV B, E
59	LD E, C	MOV E, C	MOV C, E
5A	LD E, D	MOV E, D	MOV D, E
5B	LD E, E	MOV E, E	MOV E, E
5C	LD E, H	MOV E, H	MOV H, E
5D	LD E, L	MOV E, L	MOV L, E
5E	LD E, (HL)	MOV E, M	LDR E, M
5F	LD E, A	MOV E, A	MOV A, E

(次ページへ続く)

16進	ザイログZ-80	8080	ソードZ-80
60	LD H, B	MOV H, B	MOV B, H
61	LD H, C	MOV H, C	MOV C, H
62	LD H, D	MOV H, D	MOV D, H
63	LD H, E	MOV H, E	MOV E, H
64	LD H, H	MOV H, H	MOV H, H
65	LD H, L	MOV H, L	MOV L, H
66	LD H, (HL)	MOV H, M	LDR H, M
67	LD H, A	MOV H, A	MOV A, H
68	LD L, B	MOV L, B	MOV B, L
69	LD L, C	MOV L, C	MOV C, L
6A	LD L, D	MOV L, D	MOV D, L
6B	LD L, E	MOV L, E	MOV E, L
6C	LD L, H	MOV L, H	MOV H, L
6D	LD L, L	MOV L, L	MOL L, L
6E	LD L, (HL)	MOV L, M	LDR L, M
6F	LD L, A	MOV L, A	MOV A, L
70	LD(HL), B	MOV M, B	STR B, M
71	LD (HL), C	MOV M, C	STR C, M
72	LD (HL), D	MOV M, D	STR D, M
73	LD (HL), E	MOV M, E	STR E, M
74	LD (HL), H	MOV M, H	STR H, M
75	LD (HL), L	MOV M, L	STR L, M
76	HALT	HLT	HALT
77	LD M, A	MOV M, A	STR A, M
78	LD A, B	MOV A, B	MOV B, A
79	LD A, C	MOV A, C	MOV C, A
7A	LD A, D	MOV A, D	MOV D, A
7B	LD A, E	MOV A, E	MOV E, A
7C	LD A, H	MOV A, H	MOV H, A
7D	LD A, L	MOV A, L	MOV L, A
7E	LD A, (HL)	MOV. A, M	LDR A, M
7F	LD A, A	MOV A, A	MOV A, A
80	ADD A, B	ADD B	ADD B
81	ADD A, C	ADD C	ADD C
82	ADD A, D	ADD D	ADD D
83	ADD A, E	ADD E	ADD E
84	ADD A, H	ADD H	ADD H
85	ADD A, L	ADD L	ADD L
86	ADD A (HL)	ADD M	ADD M
87	ADD A, A	ADD A	ADD A
88	ADC A, B	ADC B	ADC B
89	ADC A, C	ADC C	ADC C
8A	ADC A, D	ADC D	ADC D
8B	ADC A, E	ADC E	ADC E
8C	ADC A, H	ADC H	ADC H
8D	ADC A, (HL)	ADC L	ADC L
8E	ADC A, (HL)	ADC M	ADC M
8F	ADC A, A	ADC A	ADC A

(次ページへ続く)

(1) 16進対応表 ①

16進	ザイログZ-80	8080	ソードZ-80
9 0	SUB B	SUB B	SUB B
9 1	SUB C	SUB C	SUB C
9 2	SUB D	SUB D	SUB D
9 3	SUB E	SUB E	SUB E
9 4	SUB H	SUB H	SUB H
9 5	SUB L	SUB L	SUB L
9 6	SUB (HL)	SUB M	SUB M
9 7	SUB A	SUB A	SUB A
9 8	SBC A, B	SBB B	SBB B
9 9	SBC A, C	SBB C	SBB C
9 A	SBC A, D	SBB D	SBB D
9 B	SBC A, E	SBB E	SBB E
9 C	SBC A, H	SBB H	SBB H
9 D	SBC A, L	SBB L	SBB L
9 E	SBC A, (HL)	SBB M	SBB M
9 F	SBC A, A	SBB A	SBB A
A 0	AND B	ANA B	AND B
A 1	AND C	ANA C	AND C
A 2	AND D	ANA D	AND D
A 3	AND E	ANA E	AND E
A 4	AND H	ANA H	AND H
A 5	AND L	ANA L	AND L
A 6	AND (HL)	ANA M	AND M
A 7	AND A	ANA A	AND A
A 8	XOR B	XRA B	XOR B
A 9	XOR C	XRA C	XOR C
A A	XOR D	XRA D	XOR D
A B	XOR E	XRA E	XOR E
A C	XOR H	XRA H	XOR H
A D	XOR L	XRA L	XOR L
A E	XOR (HL)	XRA M	XOR M
A F	XOR A	XRA A	XOR A
B 0	OR B	ORA B	OR B
B 1	OR C	ORA C	OR C
B 2	OR D	ORA D	OR D
B 3	OR E	ORA E	OR E
B 4	OR H	ORA H	OR H
B 5	OR L	ORA L	OR L
B 6	OR (HL)	ORA M	OR M
B 7	OR A	ORA A	OR A
B 8	CP B	CMP B	CMP B
B 9	CP C	CMP C	CMP C
B A	CP D	CMP D	CMP D
B B	CP E	CMP E	CMP E
B C	CP H	CMP H	CMP H
B D	CP L	CMP L	CMP L
B E	CP (HL)	CMP M	CMP M
B F	CP A	CMP A	CMP A

(次ページへ続く)

16進	ザイログZ-80	8080	ソードZ-80
C 0	RET NZ	RNZ	RNZ
C 1	POP BC	POP B	POP B
C 2	JP NG, nn	JNZ nn	JNZ nn
C 3	JP nn	JMP nn	JMP nn
C 4	CALL NZ, nn	CNZ nn	CNZ nn
C 5	PUSH BC	PUSH B	PUSH B
C 6	ADD A, n	ADI B	ADDI n
C 7	RST 0	RST 0	RST 0
C 8	RET Z	RZ	RZZ
C 9	RET	RET	RET
C A	JP Z, nn	JZ nn	JZZ nn
C B	対応表③参照	—	対応表③参照
C C	CALL Z, nn	CZ nn	CZZ nn
C D	CALL nn	CALL nn	CAL nn
C E	ADC A, n	ACI n	ADCI n
C F	RST 8H	RST 1	RST 1
D 0	RET NC	RNC	RZC
D 1	POP DE	POP D	POP D
D 2	JP NC, nn	JNC nn	JZC nn
D 3	OUT (n), A	OUT n	OUT n
D 4	CALL NC, nn	CNC nn	CZC nn
D 5	PUSH DE	PUSH D	PUSH D
D 6	SUB n	SUI n	SUBI n
D 7	RST 10H	RST 2	RST 2
D 8	RET CY	RC	RNC
D 9	EXX	—	SWAP
D A	JP CY, nn	JC	JNC nn
D B	IN A, (n)	IN n	IN n
D C	CALL CY, nn	CC nn	CNC nn
D D	I X命令	—	I X命令
D E	SBC A, n	SBI n	SBBI n
D F	RST 18H	RST 3	RST 3
E 0	RET PO	RPO	RPO
E 1	POP HL	POP H	POP H
E 2	JP PO, nn	JPO nn	JPO nn
E 3	EX (SP), HL	XTHL	XTHL
E 4	CALL PO, nn	CPO nn	CPO nn
E 5	PUSH HL	PUSH H	PUSH H
E 6	AND n	ANI n	ANDI n
E 7	RST 20H	RST 4	RST 4
E 8	RET PE	RPE	RPE
E 9	JP (HL)	PCHL	PCHL
E A	JP PE, nn	JPE nn	JPE nn
E B	EX DE, HL	XCHG	XCHG
E C	CALL PE, nn	CPE nn	CPE nn
E D	対応表②参照	—	対応表②参照
E E	XOR n	XRI n	XORI n
E F	RST 28H	RST 5	RST 5

（次ページへ続く）

(1)　16進対応表　①

16進	ザイログZ-80	8080	ソードZ-80
F 0	RET P	RP	RPS
F 1	POP AF	POP PSW	POP A
F 2	JP P, nn	JP nn	JPS nn
F 3	DI	DI	DI
F 4	CALL P, nn	CP nn	CPS nn
F 5	PUSH AF	PUSH PSW	PUSH A
F 6	OR n	ORI n	ORI n
F 7	RST 30H	RST 6	RST 6
F 8	RET M	RM	RMS
F 9	LD SP, HL	SPHL	SPHL
F A	JP M, nn	JM nn	JMS nn
F B	EI	EI	EI
F C	CALL M, nn	CM nn	CMS nn
F D	I Y命令	—	I Y命令
F E	CP n	CPI n	CMPI n
F F	RST 38H	RST 7	RST 7

(2)　16進数対応表　②（Z-80 アセンブラ　EDから始まる命令）

ED××	ザイログ	ソード	ED××	ザイログ	ソード
4 2	SBC HL, BC	DSBB B	7 3	LD(nn), SP	SDBL SP
4 3	LD(nn), BC	SDBL B, nn	7 A	ADC HL, SP	DADC SP
4 4	NEG	NEG	7 B	LD SP, (nn)	LDBL SP
4 5	RETN	RETN	A 0	LDI	MVI
4 6	I M 0	—	A 1	CPI	CPI
4 7	LD I, A	PUTI	A 2	INI	INI
4 A	ADC HL, BC	DADC B	A 3	OUTI	OUTI
4 B	LD BC, (nn)	LDBL B	A 8	LDD	MVD
4 D	RFTI	RETI	A 9	CPD	CPD
4 F	LD R, A	PUTR	A A	IND	IND
5 2	SBC HL, DE	DSBB D	A B	OUTD	OUTD
5 3	LP(nn), DE	SDBL D, nn	B 0	LDIR	MVIR
5 6	IM 1	—	B 1	CPIR	CPIR
5 A	ADC HL, DE	DADC D	B 2	INIR	INIR
5 B	LD DE, (nn)	LDBL D	B 3	OUTR	—
5 E	IM 2	—	B 8	LDDR	MVDR
6 2	SBC HL, HL	DSBB H	B 9	CPDR	CPDR
6 3	LD(nn), HL	SDBL H, nn	B A	INDR	INDR
6 7	RRD	RRD	B B	OUDR	—
6 A	ADC HL, HL	DADC H		IN r, (C)	—
6 B	LD HL, (nn)	LDBL H		(01 r 000)	
6 F	RLD	RLD		OUT (C), r	—
7 2	SBC HL, SP	DSBB SP		(01 r 001)	

(3)　16進数対応表　③（Z-80 アセンブラ　CBから始まる命令）

CB××	ザイログ	ソード	CB××	ザイログ	ソード
0 0 0 0 0 r	RLC r	RLC r'	0 0 1 0 1 r	SRA r	SRA r'
0 0 0 0 1 r	RRC r	RRC r'	0 0 1 1 1 r	SRL r	SRL r'
0 0 0 1 0 r	RL r	RAL r'	0 1 b r	BIT b, r	TSTB r', b
0 0 0 1 1 r	RR r	RAR r'	1 0 b r	RES b, r	RSTB r', b
0 0 1 0 0 r	SLA r	SLL r'	1 1 b r	SET b, r	SETB r', b

- r（ザイログ）は，B，C，D，E，H，L，A，(HL)，(I X＋d)，(I Y＋d)
- r'（ソード）は，B，C，D，E，H，L，A，M，I X.d，I Y.d

付録 3　1台のフロッピーディスク装置による FDコピープログラム

（118ページの演習問題の解答）

```
E  SEQ.  LOC.  OBJ          SOURCE STATEMENT

                            ;****************************************
                            ;   FD COPY PROGRAM
                            ;****************************************
         FA23               CONOT    EQU OFA23H
         F9BC               CONIN    EQU OF9BCH
         00F8               DISK     EQU OF8H
         00F8               DCOM     EQU DISK
         00F8               DSTAT    EQU DISK
         00FA               SECTP    EQU DISK+2
         00FB               DDATA    EQU DISK+3
         00FC               WAIT     EQU DISK+4
         008C               READ     EQU 8CH
         00AC               WRITE    EQU 0ACH
         0014               SEEK     EQU 14H
         007E               BUSSY    EQU 7EH
                                     ORG 100H
         0100  310002                LD SP,200H
         0103  1601         START:   LD D,1
         0105  0600                  LD B,0 ;TR 0
         0107  3E0A                  LD A,0AH ; LINE FEED
         0109  CD23FA                CALL CONOT
         010C  3E0D                  LD A,0DH; CR
         010E  CD23FA                CALL CONOT
         0111               FDC01:
         0111  1E00                  LD E,0; READ MODE
         0113  CD2901                CALL REWR
         0116  1E01                  LD E,1
         0118  CD2901                CALL REWR
         011B  78                    LD A,B
         011C  C610                  ADD A,16
         011E  47                    LD B,A
         011F  14                    INC D
         0120  7A                    LD A,D
         0121  FE06                  CP 6
         0123  C21101                JP NZ,FDC01
         0126  C30301                JP START
                            ;----------------------------------------
                            ;     SUB READ WRITE
         0129               REWR:
         0129  F5                    PUSH AF
         012A  C5                    PUSH BC
         012B  D5                    PUSH DE
         012C  210004                LD HL,400H
         012F  1600                  LD D,0
         0131  7B                    LD A,E
         0132  FE01                  CP 1
         0134  CA3C01                JP Z,WDISP
         0137  3E52         RDISP:   LD A,'R'
         0139  C33E01                JP REWR2
         013C  3E57         WDISP:   LD A,'W'
         013E  CD23FA       REWR2:   CALL CONOT
         0141  CDBCF9                CALL CONIN
         0144  0E01         REWR3:   LD C,1
         0146  7B           REWR4:   LD A,E
         0147  FE01                  CP 1
         0149  CA5201                JP Z, WRITES
         014C  CD6E01       READS:   CALL READSEC
         014F  C35501                JP REWR5
```

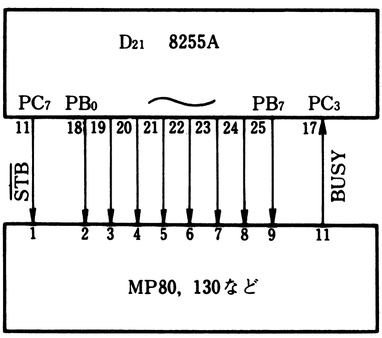
```
0152  CD9501   WRITES: CALL WRITSE
0155  0C       REWR5:  INC C
0156  79               LD A,C
0157  FE1B             CP 27
0159  DA4601           JP C,REWR4
015C  04               INC B
015D  78               LD A,B
015E  FE4D             CP 77
0160  CA6A01           JP Z,REWR6
0163  14               INC D
0164  7A               LD A,D
0165  FE10             CP 16
0167  C24401           JP NZ,REWR3
016A  D1       REWR6:  POP DE
016B  C1               POP BC
016C  F1               POP AF
016D  C9               RET
               ;--------------------------------------------
               ;   SUB READSEC
016E           READSEC:
016E  78               LD A,B
016F  D3FB             OUT (DDATA),A
0171  DBF8     BUSY:   IN A,(DSTAT)
0173  CB47             BIT 0,A
0175  20FA             JR NZ,BUSY
0177  3E14             LD A,SEEK
0179  D3F8             OUT (DCOM),A
017B  DB7E     BUSSYZ: IN A,(BUSSY)
017D  CB4F             BIT 1,A
017F  20FA             JR NZ,BUSSYZ
0181  79               LD A,C
0182  D3FA             OUT (SECTP),A
0184  3E8C             LD A,READ
0186  D3F8     READE:  OUT (DCOM),A
0188  DBFC     RLOOP:  IN A,(WAIT)
018A  B7               OR A
018B  F29401           JP P,RDDONE
018E  DBFB             IN A,(DDATA)
0190  77               LD (HL),A
0191  23               INC HL
0192  18F4             JR RLOOP
0194  C9       RDDONE: RET
               ;--------------------------------------------
               ;  SUB WRITESEC
0195  78       WRITSE: LD A,B
0196  D3FB             OUT (DDATA),A
0198  DBF8     WBUSY:  IN A,(DSTAT)
019A  CB47             BIT 0,A
019C  20FA             JR NZ,WBUSY
019E  3E14             LD A,SEEK
01A0  D3F8             OUT (DCOM),A
01A2  DB7E     WBUSSY: IN A,(BUSSY)
01A4  CB4F             BIT 1,A
01A6  20FA             JR NZ,WBUSSY
01A8  79               LD A,C
01A9  D3FA             OUT (SECTP),A
01AB  3EAC             LD A,WRITE
01AD  D3F8             OUT (DCOM),A
01AF  DBFC     WLOOP:  IN A,(WAIT)
01B1  B7               OR A
01B2  F2BB01           JP P,WDONE
01B5  7E               LD A,(HL)
01B6  D3FB             OUT(DDATA),A
01B8  23               INC HL
01B9  18F4             JR WLOOP
01BB  C9       WDONE:  RET
```

付録 4　プリンタ接続の注意点

プリンタは，機種により制御信号仕様が異なるため，各装置に応じた変更が必要となります。プリンタを正常に動作させるための方法について示します。

(1)　**EPSON・MP シリーズ**

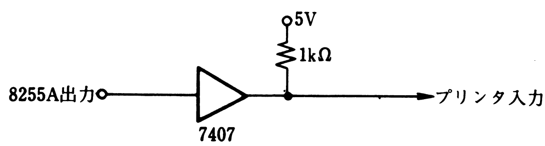
この機種に対しては，**図A**に示すように配線します。



図A　MPシリーズのプリンタに対する配線

(2)　**精工舎・GP-80，他**

GP-80やその他のプリンタも図Aに示した構成で動作すると考えられますが，もし作動しない場合は，8255 Aの出力のドライブ能力不足が考えられます。このときは，**図B**のように8255 Aからプリンタへの信号（$\overline{\text{STB}}$およびデータ）の間にバッファ(TTL 7407, オープンコレクタ)を挿入すればいいでしょう。



図B　オープンコレクタの挿入による
ドライブ能力の強化

付録5　紙テープリーダ，パンチャのパッチ方法

CP/MのBIOSは，入出力装置として紙テープリーダとパンチャが接続できるように構成されています。

リーダ入力は，他の計算機やCP/Mシステムからのデータの入力用として実際には活用されますので，実装しておくと便利なシステムです。紙テープリーダ，パンチャの回路例とBIOSへのパッチ方法について示します。

制御回路は，図4.6（第1部第4章）に示したキーボード制御用と同様に8255Aを用います。制御方法としては，**表A**に示すように，

$\overline{CS} = \overline{8}$

と設定し，PAをリーダ入力用，PBをパンチャ出力用，PCを信号制御用として用います。

この制御プログラムは，図9.15（第2部第9章）に示したセントロニクス準拠のLIST制御プログラムと同様に，$\overline{STB}$とBUSY信号を用いて構成してあります。リーダプログラムとパンチャプログラムのダンプリストを**付録6**に示してあります。

リーダのプログラムは，FD 20～FD 50，パンチャプログラムはFD 51～FD 6Eに配置してあります。ただし，このプログラムは，あくまで参考のための例ですから，購入したパンチャやリーダの仕様に合わせたプログラムを作製して挿入して下さい。

なおFD 16～FD 89は，このリーダ，パンチャ用の8255Aのモード設定部であり，ブートと同時にモード設定を行なっています。また，BIOSのアセンブラリスト中にも，次の2行，

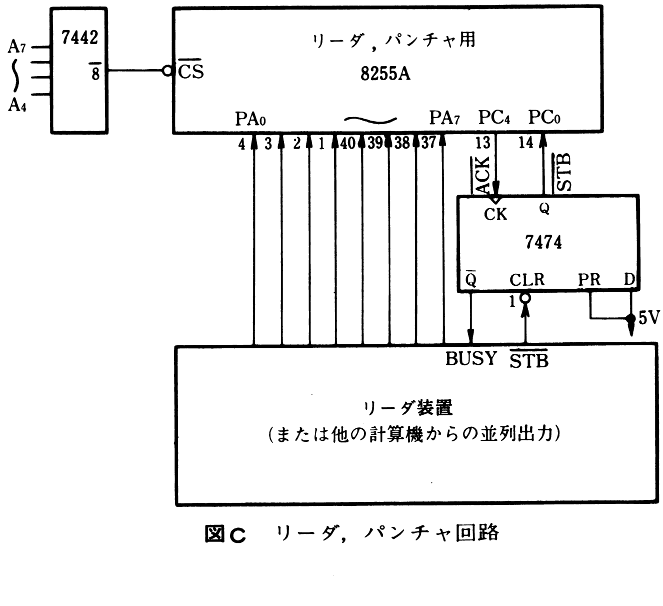
```
PUNCH ： JP    0FD51H
READER ： JP    0FD20H
```

を追加することが必要です。

リーダ回路が作動しない原因として，他部装置からの$\overline{STB}$信号の検出ミスがあります。このミスを防止するためには，**図C**のフリップフロップ回路を挿入すると良い結果が得られます。

表A　リーダ，パンチャ接続方法

8255A $\overline{CS}$	リーダ入力 $\overline{STB}$ $\overline{ACK}$	パンチ出力 $\overline{STB}$ $\overline{ACK}$
$\overline{8}$	PA_0 ～ PA_7　PC_0　PC_4	PB_0 ～ PB_7　PC_6　PC_2



図C　リーダ，パンチャ回路

付録6 CP／Mモニタのダンプリスト

本書で作製したCP/M手作りマイコンで，最終的に利用するCP/Mモニタの全ダンプリストを示します。手作りマイコンのCPU基板上にあるEPROMには，ダンプリストに従ったプログラムが書き込まれます。

```
F800 31 00 F0 21 00 00 11 00 F8 01 00 08 ED B0 C3 11
F810 F8 3E 91 D3 7F 3E 50 D3 7E CD 40 F8 3E 3E C3 EA
F820 FC CD D2 F9 FE 4D CA 4C F8 FE 50 CA 53 F9 FE 47
F830 CA B1 F9 FE 57 CA 0C FB FE 52 CA 4A FB C3 1C F8
F840 CD D9 F9 CD FE F9 3E 00 32 00 F5 C9 21 C3 F8 CD
F850 0D FA CD CB F8 CD 17 FA CD F6 F8 CD BC F9 FE 20
F860 CA 9C F8 FE 54 CA 1C F8 FE 58 CA AF F8 FE 47 D2
F870 B3 F8 FE 2F DA B3 F8 32 CE F0 CB 77 CA 81 F8 C6
F880 09 ED 6F CD BC F9 FE 47 D2 BB F8 FE 2F DA BB F8
F890 32 CF F0 CB 77 CA 9A F8 C6 09 ED 6F 23 E5 01 0D
F8A0 00 09 CD 10 F9 21 C8 F0 CD E2 FA E1 C3 5B F8 2B
F8B0 C3 55 F8 3E 3F 32 CE F0 C3 5B F8 3E 3F 32 CF F0
F8C0 C3 83 F8 0A 0D 41 44 52 0A 0D 00 D5 21 03 F5 CD
F8D0 E9 F8 ED 6F CD E9 F8 ED 6F 56 23 CD E9 F8 ED 6F
F8E0 CD E9 F8 ED 6F 6E 62 D1 C9 CD D2 F9 CB 77 CA F3
F8F0 F8 C6 09 E6 0F C9 E5 C5 F5 01 05 00 ED 42 06 13
F900 CD 10 F9 23 10 FA 21 C8 F0 CD E2 FA F1 C1 E1 C9
F910 3E 20 CD 23 FA CD 25 F9 3E 20 CD 23 FA 7E CD 2E
F920 F9 CD 17 FA C9 7C CD 2E F9 7D CD 2E F9 C9 E5 21
F930 02 F5 77 ED 6F CD 3F F9 ED 6F CD 3F F9 E1 C9 E6
F940 0F FE 0A 30 04 C6 30 18 06 D6 09 E6 07 C6 40 CD
F950 23 FA C9 21 00 10 CD D2 F9 FE 57 CA 8C F9 3E 82
F960 D3 0F 3E E0 D3 0E 3E C0 D3 0E 3E E0 D3 0E 16 FF
F970 CD FF FA 7D D3 0C 7C E6 0F C6 E0 D3 0E DB 0D 77
F980 CD 10 F9 23 7C FE 18 20 EA C3 1C F8 3E 80 D3 0F
F990 3E 70 D3 0E 7E D3 0D 7D D3 0C 7C E6 0F C6 50 D3
F9A0 0E CD 10 F9 3E 70 D3 0E 23 7C FE 18 20 E6 C3 1C
F9B0 F8 21 C3 F8 CD 0D FA CD CB F8 E5 C9 C5 DB 7E CB
F9C0 47 20 FA DB 7C 00 E6 7F 47 DB 7E CB 47 28 FA 78
F9D0 C1 C9 CD BC F9 CD 23 FA C9 3E 00 47 DD 21 F0 F9
F9E0 D3 F6 DD 7E 00 D3 F7 DD 23 04 78 FE 0E 38 F1 C9
F9F0 40 32 37 04 18 01 14 16 00 0B 65 08 00 00 E5 21
FA00 00 F0 3E 20 77 7C 23 FE F5 20 F7 E1 C9 7E B7 C8
FA10 4F CD 23 FA 23 18 F6 E5 21 20 FA CD 0D FA E1 C9
FA20 0A 0D 00 F5 C5 D5 E5 32 01 F5 FE 0D 28 38 FE 0A
FA30 28 41 FE 7F 28 14 CD BB FA 3A 00 F5 FE 31 CA AB
FA40 FA 3C 32 00 F5 E1 D1 C1 F1 C9 3A 00 F5 B7 28 16
FA50 3A 00 F5 3D 32 00 F5 3E 20 32 01 F5 CD BB FA 2B
FA60 CD E2 FA C3 45 FA 3E 00 32 00 F5 21 B6 F3 CD E2
```

〔F800〜FB0B〕
図7.19 モニタプログラムの部分

FA70	FA	18	D2	CD	78	FA	18	CD	DD	21	32	F0	21	00	F0	DD	
FA80	7E	00	CD	D3	FA	77	23	DD	23	7C	FE	F4	20	F1	7D	FE	
FA90	40	20	EC	DB	7E	CB	47	20	0C	DB	7C	2F	E6	7F	FE	13	
FAA0	20	03	CD	BC	F9	16	20	CD	FF	FA	C9	CD	78	FA	3E	00	
FAB0	32	00	F5	21	B6	F3	CD	E2	FA	18	8A	C5	21	B6	F3	3A	
FAC0	00	F5	4F	06	00	09	3A	01	F5	CD	D3	FA	77	23	CD	E2	
FAD0	FA	C1	C9	F5	DB	7E	CB	57	28	FA	DB	7E	CB	57	20	FA	
FAE0	F1	C9	3E	0E	CD	D3	FA	D3	F6	7C	E6	0F	CD	D3	FA	D3	
FAF0	F7	3E	0F	CD	D3	FA	D3	F6	7D	CD	D3	FA	D3	F7	C9	C5	
FB00	F5	01	00	00	03	78	BA	20	FB	F1	C1	C9	21	C3	F8	CD	〔FB0C～FCC2〕図8.18　CP／Mモニタプログラムの部分
FB10	0D	FA	CD	CB	F8	22	C3	F4	21	42	FB	CD	0D	FA	CD	80	
FB20	FB	4F	CD	A9	FB	1E	00	01	80	00	1C	7B	32	C1	F5	CD	
FB30	90	FC	2A	C3	F4	09	22	C3	F4	7B	FE	10	C2	2A	FB	C3	
FB40	1C	F8	0A	0D	54	52	4B	0A	0D	00	21	C3	F8	CD	0D	FA	
FB50	CD	CB	F8	22	C3	F4	21	42	FB	CD	0D	FA	CD	80	FB	4F	
FB60	CD	A9	FB	1E	00	01	80	00	1C	7B	32	C1	F5	CD	12	FC	
FB70	2A	C3	F4	09	22	C3	F4	7B	FE	10	C2	68	FB	C3	1C	F8	
FB80	C5	D5	16	00	CD	BC	F9	FE	0D	28	13	CD	23	FA	E6	0F	
FB90	47	7A	CB	27	4F	CB	27	CB	27	81	80	57	18	E6	7A	D1	
FBA0	C1	C9	0E	00	3E	FF	32	D0	F5	79	32	C0	F5	C5	47	3E	
FBB0	0A	32	FE	F5	DB	F9	D5	16	01	CD	FF	FA	D1	B8	20	02	
FBC0	C1	C9	D5	16	01	CD	FF	FA	D1	78	D3	FB	DB	F8	CB	47	
FBD0	20	FA	3E	14	D3	F8	DB	7E	CB	4F	20	FA	DB	F8	E6	91	
FBE0	28	DE	3A	FE	F5	3D	20	C9	C1	CD	3A	FC	18	BF	79	32	
FBF0	C1	F5	C9	3A	D0	F5	B7	28	11	2F	32	D0	F5	DB	F9	D3	
FC00	FB	3E	14	D3	F8	DB	FC	3E	04	C9	DB	F8	E6	20	28	F7	
FC10	AF	C9	3E	0A	CD	7B	FC	C6	88	D3	F8	DB	FC	B7	F2	27	
FC20	FC	DB	FB	77	23	18	F4	DB	F8	E6	9D	C8	CD	49	FC	3A	
FC30	FD	F5	3D	20	DF	CD	3A	FC	18	D8	21	E1	FC	CD	0D	FA	
FC40	CD	BC	F9	FE	03	C0	C3	F3	5A	57	E6	10	20	03	7A	B7	
FC50	C9	3E	C4	D3	F8	DB	FC	DB	FB	47	DB	FC	B7	CB	7F	28	
FC60	04	DB	FB	18	F5	DB	F8	B7	28	05	CD	A2	FB	18	03	78	
FC70	D3	F9	3A	C0	F5	CD	AD	FB	7A	B7	C9	32	FD	F5	3E	D0	
FC80	D3	F8	E3	E3	2A	C3	F4	3A	C1	F5	D3	FA	CD	F3	FB	C9	
FC90	3E	0A	CD	7B	FC	C6	A8	D3	F8	DB	FC	B7	F2	A5	FC	7E	〔FCC3～FCE9〕図9.15　LISTプログラムの部分
FCA0	D3	FB	23	18	F4	DB	F8	E6	FD	C8	3A	49	24	3D	CD	3A	
FCB0	FC	C3	90	FC	60	69	22	C3	F4	C9	DB	7E	CB	47	3E	FF	
FCC0	C8	2F	C9	F5	DB	7E	CB	5F	20	FA	79	D3	7D	3E	50	D3	〔FCEA～FD1F〕図9.11　ブートストラップローダプログラムの部分
FCD0	7E	3E	10	D3	7E	3E	50	D3	7E	DB	7E	CB	5F	28	FA	F1	
FCE0	C9	45	52	52	4F	52	2E	30	20	00	21	00	00	01	01	00	
FCF0	78	D3	FB	DB	F8	CB	47	20	FA	3E	14	D3	F8	DB	7E	CB	
FD00	4F	20	FA	79	D3	FA	3E	8C	D3	F8	DB	FC	B7	F2	16	FD	
FD10	DB	FB	77	23	18	F4	3E*	91*	D3*	8F*	C3	00	00	00	00	00	〔FD20～FD50〕付録5　リーダプログラムの部分
FD20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
FD30	00	DB	8E	CB	47	20	FA	DB	8C	32	04	F5	3E	EF	D3	8E	
FD40	F5	F1	3E	FF	D3	8E	D5	16	01	CD	FF	FA	D1	3A	04	F5	
FD50	C9	F5	DB	8E	CB	57	28	FA	79	D3	8D	3E	FF	D3	8E	3E	〔FD51～FD6F〕付録5　パンチャプログラムの部分
FD60	BF	D3	8E	DB	8E	CB	57	20	FA	3E	FF	D3	8E	F1	C9	00	

＊リーダ，パンチャ用の8255Aモード設定部，未使用の場合は“00”

付録7 BIOSプログラムリスト

本書で作製したCP/M手作りマイコン用にパッチされ，不必要な部分を削除したBIOSプログラムリストを以下に示します。

```
                 ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                 ; TARBELL ELECTRONICS
                 ; 2.2 VERSION OF 2-19-80
003C =           MSIZE   EQU  60              ;MEMORY SIZE IN KBYTES.
0001 =           NDISK   EQU  1               ;DEFINES THE NUMBER DRIVES IN SYSTEM.
00F8 =           DISK    EQU  0F8H            ;DISK BASE ADDRESS.
00F8 =           DCOM    EQU  DISK            ;DISK COMMAND PORT.
00F8 =           DSTAT   EQU  DISK            ;DISK STATUS PORT.
00F9 =           TRACK   EQU  DISK+1          ;DISK TRACK PORT.
00FA =           SECTP   EQU  DISK+2          ;DISK SECTOR PORT.
00FB =           DDATA   EQU  DISK+3          ;DISK DATA PORT.
00FC =           WAIT    EQU  DISK+4          ;DISK WAIT PORT.
00FC =           DCONT   EQU  DISK+4          ;DISK CONTROL PORT.
000A =           RTCNT   EQU  10              ;RETRY COUNT.
0003 =           IOBYTE  EQU  3                        ;ADDRESS OF I/O BYTE.
A000 =           CBASE   EQU  (MSIZE-20)*1024          ;BIAS FOR LARGER THAN 20K.
D400 =           CPMB    EQU  CBASE+3400H              ;START OF CPM 2.0
DC06 =           BDOS    EQU  CPMB+806H                ;START OF BDOS 2.0.
EA00 =           BIOS    EQU  CPMB+1600H               ;START OF CBIOS IO.
0004 =           CDISK   EQU  4                        ;LOCATION 4 IS CURRENT DISK.
002C =           NSECTS  EQU  44.                      ;NUMBER OF SECTORS IN IT.
EA00                     ORG  BIOS                     ;START OF CBIOS STRUCTURE.
EA00 C39CEA              JMP  BOOT                     ;FROM COLD START LOADER.
EA03 C3F1EA      WBOOTE: JMP  WBOOT                    ;FROM WARM BOOT.
EA06 C34AEB              JMP  CONST                    ;CHECK CONSOLE KB STATUS.
EA09 C34DEB              JMP  CONIN                    ;READ CONSOLE CHARACTER.
EA0C C353EB              JMP  CONOT                    ;WRITE CONSOLE CHARACTER.
EA0F C3CAEB              JMP  LIST                     ;WRITE LISTING CHAR.
EA12 C3CDEB              JMP  PUNCH                    ;WRITE PUNCH CHAR.
EA15 C3D0EB              JMP  READER                   ;READ READER CHAR.
EA18 C393EB              JMP  HOME                     ;MOVE DISK TO TRACK ZERO.
EA1B C358EB              JMP  SELDSK                   ;SELECT DISK DRIVE.
EA1E C395EB              JMP  SETTRK                   ;SEEK TO TRACK IN REG A.
EA21 C398EB              JMP  SETSEC                   ;SET SECTOR NUMBER.
EA24 C3A1EB              JMP  SETDMA                   ;SET DISK STARTING ADR.
EA27 C3A4EB              JMP  READ                     ;READ SELECTED SECTOR.
EA2A C3A7EB              JMP  WRITE                    ;WRITE SELECTED SECTOR.
EA2D C3CAEB              JMP  PRSTAT                   ;LIST STATUS CHECK.
EA30 C39BEB              JMP  SECTRAN                  ;SECTOR TRANSLATE ROUTINE.
EA33 =           DPBASE  EQU  $                        ;BASE OF DISK PARAMETER BLOCK
EA33 82EA0000    DPE0:   DW   XLT0,0000H               ;TRANSLATE TABLE
EA37 00000000            DW   0000H,0000H              ;SCRATCH AREA
EA3B E1EB73EA            DW   DIRBUF,DPB0              ;DIR BUFF, PARM BLOCK
EA3F 80EC61EC            DW   CSV0,ALV0                ;CHECK, ALLOC VECTORS
EA43 82EA0000    DPE1:   DW   XLT1,0000H
EA47 00000000            DW   0000H,0000H
EA4B E1EB73EA            DW   DIRBUF,DPB1
EA4F AFEC90EC            DW   CSV1,ALV1
EA53 82EA0000    DPE2:   DW   XLT2,0000H
EA57 00000000            DW   0000H,0000H
EA5B E1EB73EA            DW   DIRBUF,DPB2
EA5F DEECBFEC            DW   CSV2,ALV2
EA63 82EA0000    DPE3:   DW   XLT3,0000H
EA67 00000000            DW   0000H,0000H
EA6B E1EB73EA            DW   DIRBUF,DPB3
EA6F 0DEDEEEC            DW   CSV3,ALV3
EA73 =           DPB0    EQU  $                        ;ONE OF 4 DISK PARM. BLOCKS
EA73 1A00                DW   26                       ;SECTORS/TRACK
EA75 03                  DB   3                        ;BLOCK SHIFT
EA76 07                  DB   7                        ;BLOCK MASK
```

```
EA77 00                    DB    0                  ;EXTNT MASK
EA78 F200                  DW    242                ;DISK SIZE - 1
EA7A 3F00                  DW    63                 ;DIRECTORY MAX.
EA7C C0                    DB    192                ;ALLOC0
EA7D 00                    DB    0                  ;ALLOC1
EA7E 1000                  DW    16                 ;CHECK SIZE
EA80 0200                  DW    2                  ;NUMBER OF SYSTEM TRACKS
EA82 =          XLT0       EQU   $                  ;START OF TRANS. TABLE
EA82 01070D1319            DB    1,7,13,19,25
EA87 050B111703            DB    5,11,17,23,3
EA8C 090F150208            DB    9,15,21,2,8
EA91 0E141A060C            DB    14,20,26,6,12
EA96 1218040A10            DB    18,24,4,10,16,22
EA73 =          DPB1       EQU   DPB0               ;EQUIVALENT PARAMETERS
EA82 =          XLT1       EQU   XLT0               ;SAME TRANSLATE TABLE
EA73 =          DPB2       EQU   DPB0
EA82 =          XLT2       EQU   XLT0
EA73 =          DPB3       EQU   DPB0
EA82 =          XLT3       EQU   XLT0
EA9C 318000     BOOT:      LXI   SP,80H      ;SET STACK POINTER.
EA9F 00000000              DW    0,0         ;LEAVE SPACE FOR INIT.
EAA3 00000000              DW    0,0
EAA7 00000000              DW    0,0
EAAB 00000000              DW    0,0
EAAF AF                    XRA   A           ;CLEAR SCRATCH AREA.
EAB0 320300                STA   IOBYTE      ;CLEAR I/O BYTE.
EAB3 320400                STA   CDISK       ;SELECT DRIVE ZERO
EAB6 0E07                  MVI   C,ENDZ-STARTZ    ;GET LENGTH OF ZERO AREA.
EAB8 21D7EB                LXI   H,STARTZ    ;GET SCRATCH ADDRESS.
EABB 77         BOOTL:     MOV   M,A         ;PUT ZERO IN MEMORY.
EABC 23                    INX   H           ;INCREMENT POINTER.
EABD 0D                    DCR   C           ;DECREMENT COUNTER.
EABE C2BBEA                JNZ   BOOTL       ;LOOP TILL DONE.
EAC1 3EF2                  MVI   A,0F2H      ;SET LATCH CODE = F2.
EAC3 32DFEB                STA   LATCH
EAC6 CDD6EA                CALL  SETUP       ;SET UP JUMPS.
EAC9 21B4EB                LXI   H,SMSG      ;PRINT OPENING MESSAGE.
EACC CDAAEB                CALL  PMSG
EACF 3A0400     GOCPM:     LDA   CDISK       ;GET DISK NUMBER TO
EAD2 4F                    MOV   C,A         ;PASS TO CCP IN C.
EAD3 C300D4                JMP   CPMB        ;JUMP TO CCP.
EAD6 3EC3       SETUP:     MVI   A,0C3H      ;PUT JMP TO WBOOT
EAD8 320000                STA   0           ;ADR AT ZERO.
EADB 2103EA                LXI   H,WBOOTE
EADE 220100                SHLD  1
EAE1 320500                STA   5
EAE4 2106DC                LXI   H,BDOS      ;PUT JUMP TO BDOS
EAE7 220600                SHLD  6           ;AT ADR 5,6,7.
EAEA 218000                LXI   H,80H       ;SET DEFAULT DMA ADR.
EAED 22D5EB                SHLD  DMAADD
EAF0 C9                    RET               ;RETURN FROM SETUP.
EAF1 318000     WBOOT:     LXI   SP,80H      ;SET STACK POINTER.
EAF4 3A0400                LDA   CDISK       ;SAVE DISK NUMBER.
EAF7 32DEEB                STA   TEMP
EAFA 0E00                  MVI   C,0         ;SELECT DISK ZERO.
EAFC CD58EB                CALL  SELDSK
EAFF CD93EB                CALL  HOME        ;MOVE TO TRACK ZERO.
EB02 C23FEB                JNZ   RDERR       ;IF ERROR, PRINT MESSAGE.
EB05 162C                  MVI   D,NSECTS    ;GET # SECTORS FOR CPM READ.
EB07 010200                LXI   B,2         ;TRACK (B)=0, SECTOR (C)=2.
EB0A 2100D4                LXI   H,CPMB      ;GET STARTING ADDRESS.
EB0D C5         RDBLK:     PUSH  B           ;SAVE B&C.
EB0E 48                    MOV   C,B         ;GO TO TRACK IN B.
EB0F CD95EB                CALL  SETTRK
EB12 C1                    POP   B           ;RESTORE B&C.
EB13 C23FEB                JNZ   RDERR       ;IF ERROR, PRINT MESSAGE.
EB16 22C3F4     RBLK1:     SHLD  0F4C3H
EB19 CD98EB                CALL  SETSEC      ;READ STARTING AT SECTOR IN C
EB1C CDA4EB                CALL  READ
EB1F C23FEB                JNZ   RDERR       ;IF ERROR, PRINT MESSAGE.
```

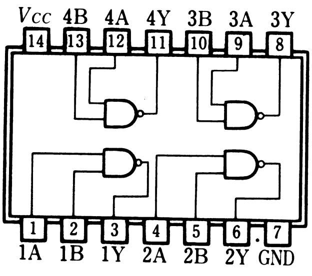
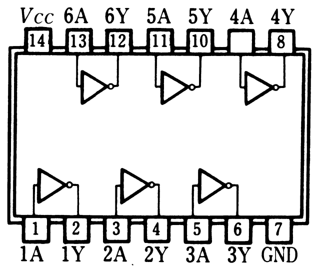
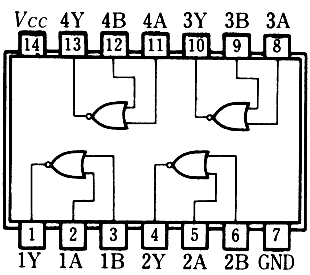
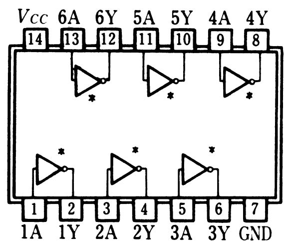
```
EB22 15                 DCR   D           ;DECREMENT SECTOR COUNT.
EB23 CA33EB             JZ    ALDON       ;ALL DONE WHEN D=0.
EB26 0C                 INR   C           ;INCREMENT SECTOR NUMBER.
EB27 79                 MOV   A,C         ;IF SECTOR NUMBER
EB28 FE1B               CPI   27          ;IS NOT 27,
EB2A DA16EB             JC    RBLK1       ;KEEP READING ON THIS TRACK.
EB2D 0E01               MVI   C,1         ;OTHERWISE, RESET SECTOR=1,
EB2F 04                 INR   B           ;INCREMENT TRACK NUMBER,
EB30 C30DEB             JMP   RDBLK       ;AND READ NEXT TRACK.
EB33 3ADEEB    ALDON:   LDA   TEMP        ;RESTORE DISK NUMBER.
EB36 320400             STA   CDISK
EB39 CDD6EA             CALL  SETUP       ;SET UP JUMPS.
EB3C C3CFEA             JMP   GOCPM       ;GO BACK TO CPM.
EB3F 0E42      RDERR:   MVI   C,'B'       ;GET BOOT ERROR CODE.
EB41 CD53EB             CALL  CONOT       ;PRINT IT.
EB44 CD4DEB             CALL  CONIN       ;READ A CHAR FROM CONSOLE.
EB47 C3F1EA             JMP   WBOOT       ;DO A WARM BOOT.
EB4A C3BAFC    CONST:   JMP 0FCBAH
EB4D CDBCF9    CONIN:   CALL 0F9BCH
EB50 E67F                ANI 7FH
EB52 C9                  RET
EB53 79        CONOT:   MOV A,C
EB54 CD23FA             CALL 0FA23H
EB57 C9                 RET
EB58 210000    SELDSK: LXI H,0                    ;SET UP FOR ERROR CODE
EB5B 79                 MOV   A,C                 ;GET NEW DRIVE.
EB5C FE01               CPI   NDISK               ;CALLING UNDEFINED DRIVE ?
EB5E D0                 RNC                       ;IF NO CY, H,L TELLS CPM YES.
EB5F 32D8EB             STA   PDISK               ;WE ARE OK, SAVE NEW DISK NUMB.
EB62 21D7EB             LXI   H,DISKNO            ;GET OLD DRIVE NUMBER.
EB65 7E        SELMOR: MOV A,M            ;GET OLD DISK NUMBER.
EB66 5F                 MOV   E,A         ;PUT OLD DISK NO. IN D&E.
EB67 1600               MVI   D,0
EB69 21DAEB             LXI   H,TRTAB     ;GET ADDRESS OF TRACK TABLE.
EB6C E5                 PUSH H            ;SAVE ADDRESS OF TRTAB.
EB6D 19                 DAD   D           ;ADD DISK NO. TO ADDRESS.
EB6E DBF9               IN    TRACK       ;READ 1771 TRACK REGISTER.
EB70 77                 MOV   M,A         ;PUT INTO TABLE.
EB71 79                 MOV   A,C         ;GET NEW DISK NUMBER.
EB72 5F                 MOV   E,A         ;PUT NEW DISK NO. IN D&E.
EB73 E1                 POP   H           ;RESTORE ADDRESS OF TRTAB.
EB74 19                 DAD   D           ;ADD DISK NO. TO ADDRESS.
EB75 7E                 MOV   A,M         ;GET NEW TRACK NUMBER.
EB76 D3F9               OUT   TRACK       ;PUT INTO 1771 TRACK REG.
EB78 79                 MOV   A,C         ;UPDATE OLD DISK NUMBER.
EB79 32D7EB             STA   DISKNO
EB7C 2F                 CMA               ;BITS INVERTED INTO LATCH.
EB7D 87                 ADD   A           ;PUT BITS 1&2 AT 4&5.
EB7E 87                 ADD   A
EB7F 87                 ADD   A
EB80 87                 ADD   A
EB81 F602               ORI   2           ;MAKE LATCH COMMAND.
EB83 32DFEB             STA   LATCH       ;SAVE NEW LATCH CODE.
EB86 2AD8EB             LHLD  PDISK               ;LOAD DISK NUMBER AND ZERO BYTE
EB89 1133EA             LXI   D,DPBASE            ;POINT TO DISK PARM START.
EB8C 29                 DAD   H                   ;*2
EB8D 29                 DAD   H                   ; *4
EB8E 29                 DAD   H                   ;   *8
EB8F 29                 DAD   H                   ;     *16
EB90 19                 DAD   D                   ;COMPUTE INDEX FOR THE DRIVE
EB91 AF                 XRA   A           ;SET A = 0.
EB92 C9                 RET               ;RETURN FROM SELDSK.
EB93 0E00      HOME:    MVI   C,0         ;SEEK TO TRACK ZERO.
EB95 C3A9FB    SETTRK: JMP 0FBA9H
EB98 C3EEFB    SETSEC: JMP 0FBEEH
EB9B EB        SECTRAN:           XCHG    ;H,L=TRANS
EB9C 09                 DAD   B                   ;H,L = TRANS (SECTOR)
EB9D 6E                 MOV   L,M                 ;L   = TRANS (SECTOR)
EB9E 2600               MVI   H,0                 ;CLEAR REG H
EBA0 C9                 RET                       ;H,L = TRANSLATED SECTOR
```

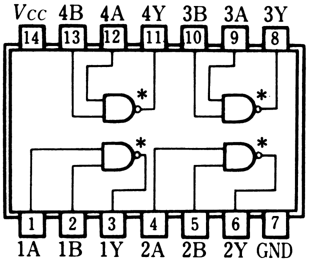
```
EBA1 C3B4FC      SETDMA: JMP 0FCB4H
EBA4 C312FC      READ:   JMP 0FC12H
EBA7 C390FC      WRITE:  JMP 0FC90H
EBAA C30DFA      PMSG:   JMP 0FA0DH
EBAD 4552524F52ERRMSG: DB   'ERROR.',0
EBB4 0D0A546172SMSG:   DB   0DH,0AH,'Tarbell '
EBBE 3630                DB   MSIZE/10+'0',MSIZE MOD 10 + '0'
EBC0 4B2043504D          DB   'K CPM 2.2',0
                 PRSTAT:
EBCA C3C3FC      LIST:   JMP 0FCC3H
EBCD C330FD      PUNCH:  JMP 0FD30H
EBD0 C351FD      READER: JMP 0FD51H
EBD3             TRK:    DS   1           ;CURRENT TRACK NUMBER.
EBD4             SECT:   DS   1           ;CURRENT SECTOR NUMBER.
EBD5             DMAADD: DS   2           ;DISK TRANSFER ADDRESS.
                 STARTZ:                  ;START OF ZEROED AREA.
EBD7             DISKNO: DS   1           ;DISK NUMBER
EBD8             PDISK:  DS   2           ;NEW DISK TO SELECT AND A ZERO BYTE
EBDA             TRTAB:  DS   4
                 ENDZ:                    ;END OF ZEROED AREA.
EBDE             TEMP:   DS   1           ;TEMPORARY STORAGE.
EBDF             LATCH:  DS   1           ;NEW CODE FOR LATCH.
EBE0             CLATCH: DS   1           ;CURRENT CODE IN LATCH.
EBE1 =           BEGDAT  EQU  $
EBE1             DIRBUF: DS   128         ;DIRECTORY BUFFER
EC61             ALV0:   DS   31
EC80             CSV0:   DS   16
EC90             ALV1:   DS   31
ECAF             CSV1:   DS   16
ECBF             ALV2:   DS   31
ECDE             CSV2:   DS   16
ECEE             ALV3:   DS   31
ED0D             CSV3:   DS   16
ED1D =           ENDDAT  EQU  $
013C =           DATSIZ  EQU  $-BEGDAT              ;TOTAL SIZE OF DISK PARM STORAGE
ED1D                      END
```

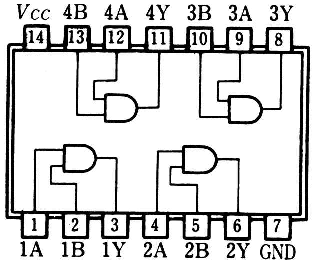
付録8　手作りマイコンに使用する　ICとピン配置一覧

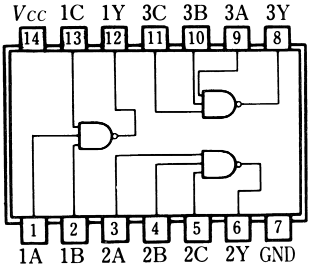
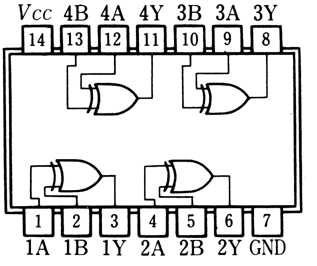
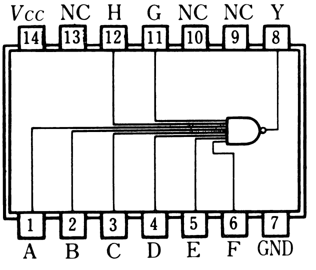
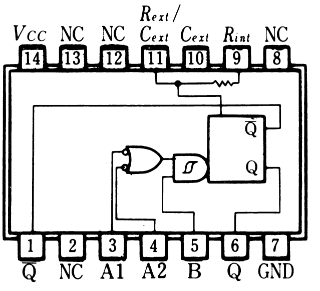
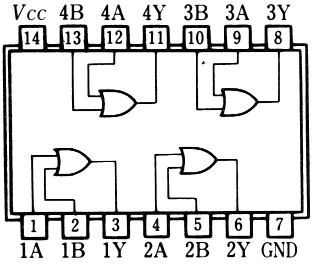
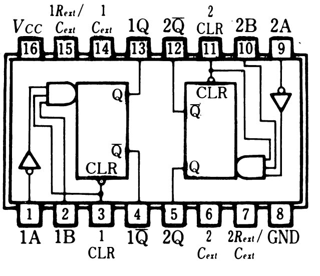
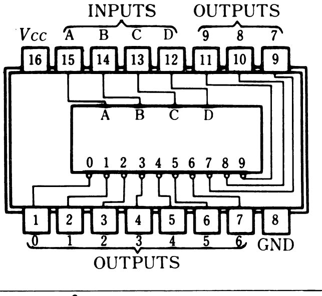
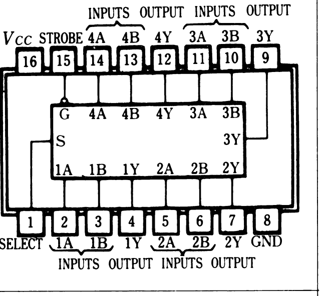
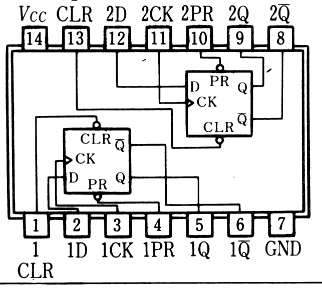
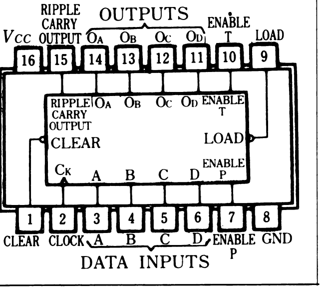
番　　　　号	機　　　　能	個　数	ピ　ン　配　置
Z－80A	マイクロプロセッサ	1	28ページ参照
8255A	インターフェース	2	34ページ参照
HD 46505 A	CRTコントロール用IC	1	43ページ参照
1771	FDコントロール用IC	1	108ページ参照
2114（450 ns でも可）	スタティック RAM	2	47ページ参照
2716	2K EPROM	4	32ページ参照
4164	64K DRAM	8	52ページ参照
MCM 66740 または EPROM 2716	キャラクタジェネレータ	1	47ページ参照
TL 497 ACN	DC－DCコンバータ	1	179ページ参照

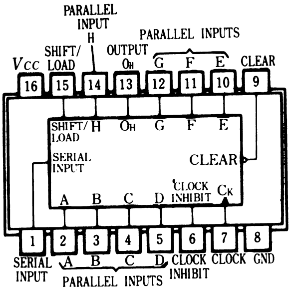
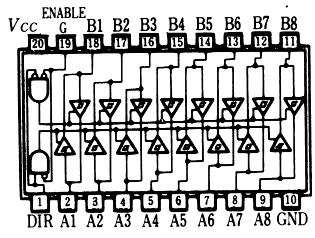
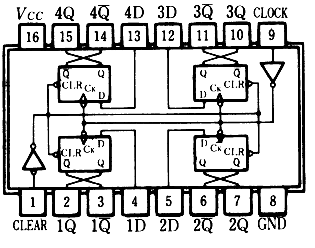
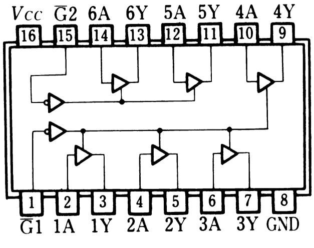
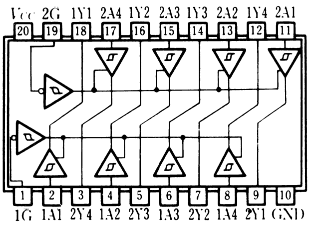
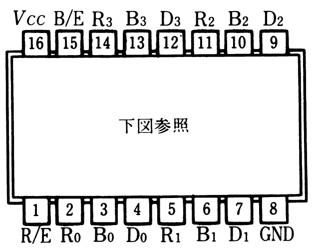
（TTL IC）

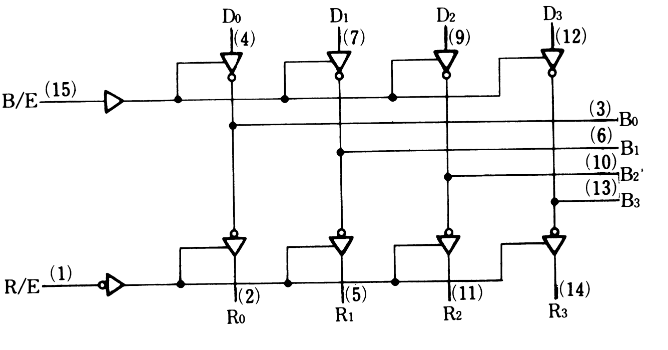
<table>
<tr><th>番　号</th><th>個数</th><th>ピン配置</th><th>番　号</th><th>個数</th><th>ピン配置</th></tr>
<tr><td>7400
74LS 00</td><td>7</td><td>
</td><td>74LS 04</td><td>5</td><td>
</td></tr>
<tr><td>74LS 02</td><td>1</td><td rowspan="2">
</td><td rowspan="2">7406</td><td rowspan="2">1</td><td rowspan="2">

＊オープンコレクタ</td></tr>
<tr><td>7402</td><td>1</td></tr>
<tr><td>74LS 03</td><td>1</td><td>

＊オープンコレクタ</td><td>74CS 08</td><td>1</td><td>
</td></tr>
</table>

番　　号	個数	ピン配置	番　　号	個数	ピン配置
7410	1		74LS 86	4	
74LS 30	1		74121	1	
74LS 32	3		74LS 123	3	
74LS 42	5		74LS 157	5	
74LS 74	4		74LS 163	1	

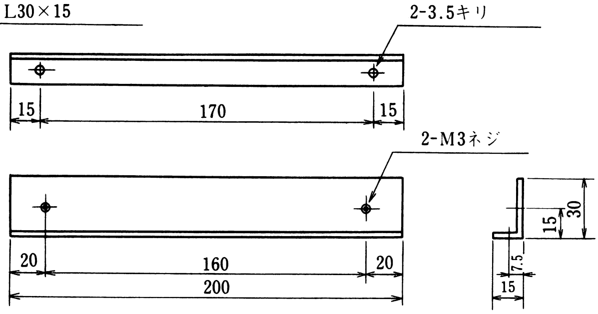
番　　号	個数	ピ　ン　配　置	番　　号	個数	ピ　ン　配　置
74LS 166	3		74LS 245	2	
74LS 175	1		74LS 367	4	
74LS 244	2		8 T26	2	



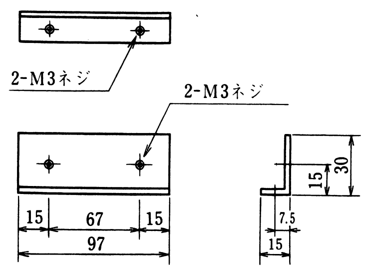
8T26の内部回路

付録9 手作りケースの製作図面

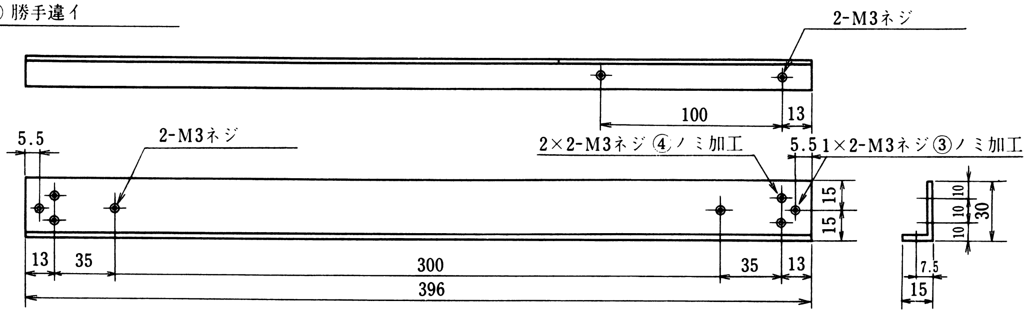
① L30×15



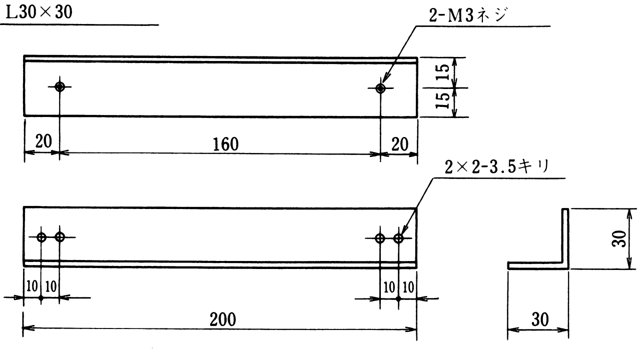
② L30×15



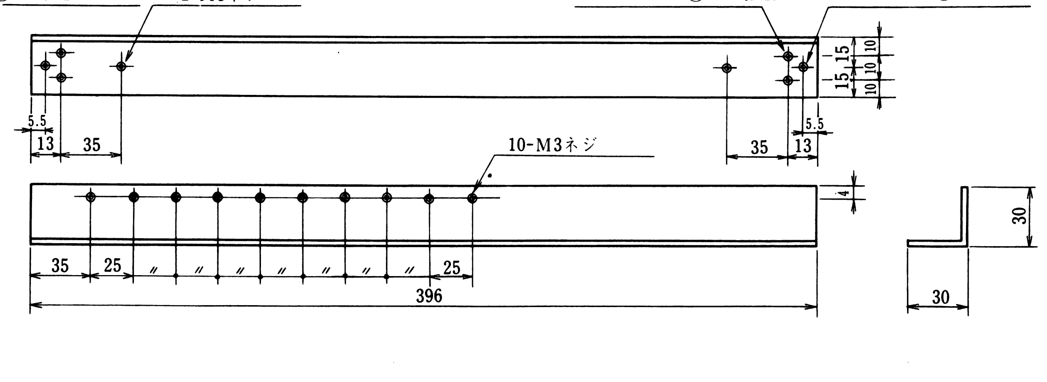
③ L30×15
④ 勝手違イ



⑤ L30×30

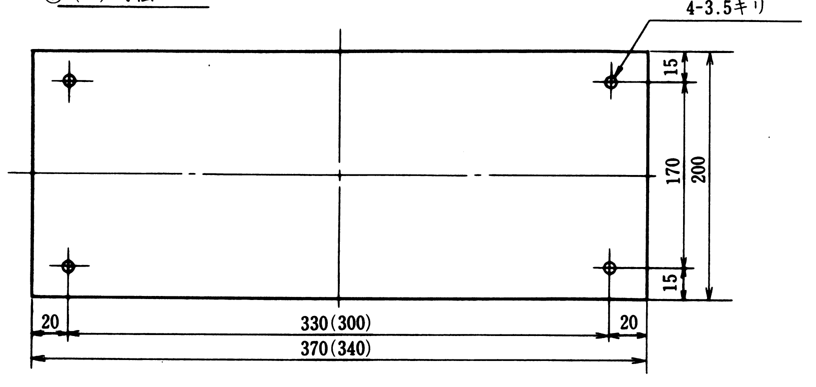


⑥ L30×30
⑦ 勝手違イ

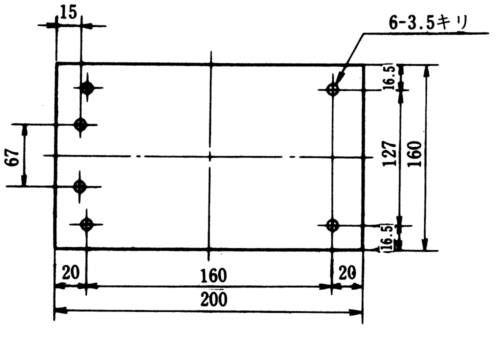


丸小ネジ～M3×ℓ6-40本（フレーム用）
〃　～M3×ℓ8-20本（基盤ソケット用）

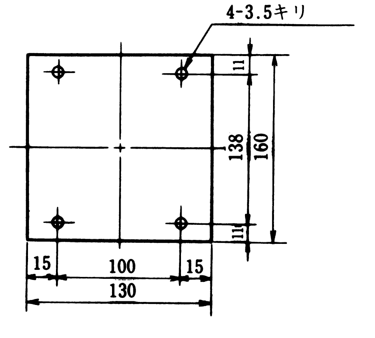
⑧ t＝2
⑨（ ）寸法



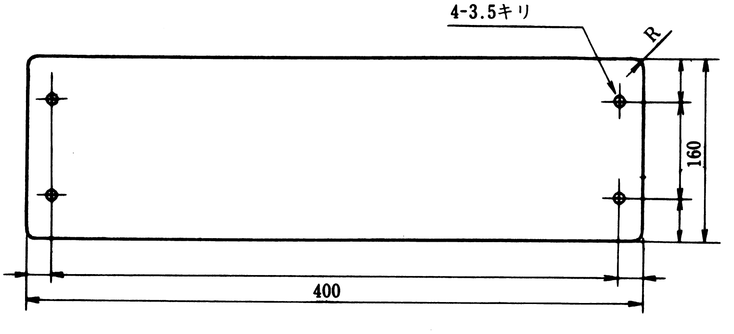
⑩ t＝2



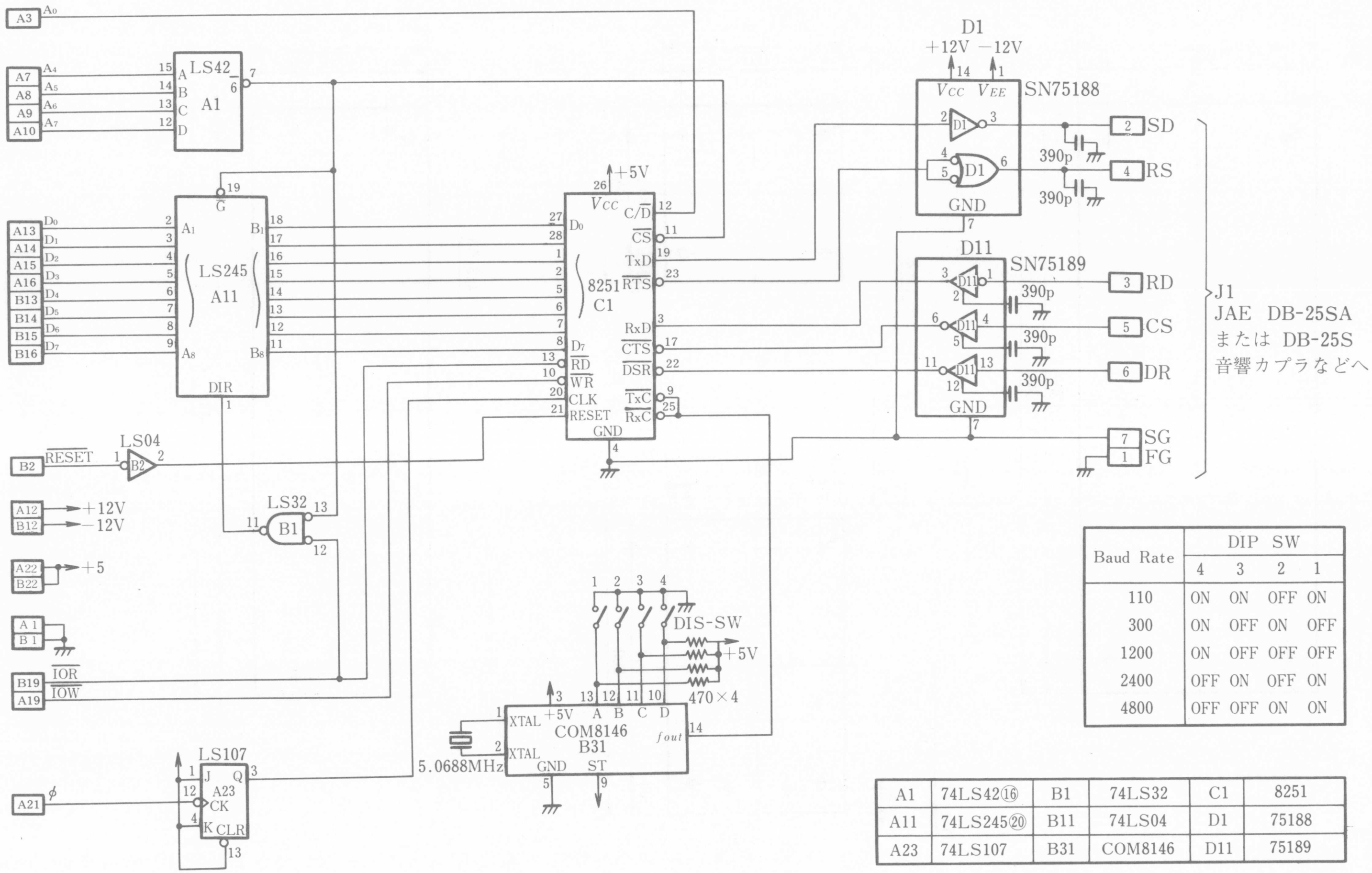
⑪ t＝2



⑫ t＝2



付録10　RS232Cインターフェース基板の配線図



Baud Rate	DIP SW 4	3	2	1
110	ON	ON	OFF	ON
300	ON	OFF	ON	OFF
1200	ON	OFF	OFF	OFF
2400	OFF	ON	OFF	ON
4800	OFF	OFF	ON	ON

A1	74LS42⑯	B1	74LS32	C1	8251
A11	74LS245⑳	B11	74LS04	D1	75188
A23	74LS107	B31	COM8146	D11	75189

参考文献

第2章

2－1） 寺田監訳：Z－80マイクロプロセッサ，丸善

2－2） 横田：Z－80の使い方，オーム社

2－3） Z－80テクニカルマニュアル，エレクトロニクスダイジェスト社

2－4） MIL-STD-806B "GRAPHIC SYMBOLS FOR LOGIC DIAGRAMS"

第4章

4－1） マイクロコンピュータユーザーズマニュアルMCS－48，インテル

4－2） 日立マイクロコンピュータシステムCRTコントローラHD46505ユーザーズマニュアル，日立製作所

4－3） 岡田正：64K DRAMの使い方とメモリシステムの設計，トランジスタ技術，1981年10月号

4－4） 猪飼國夫：メモリを自在に使うには，トランジスタ技術，1982年5月号

4－5） 松本吉彦：Z－80CPUとメモリ，トランジスタ技術，1980年11月号

4－6） 新里：テキストの送信と受信を可能にするTSS端末プログラム，東大センターニュース，Vol. 14，No，6，1982，p. 30-41

第7章

7－1） 横山英一：CRTディスプレイの設計，トランジスタ技術，1981年6月号

第8章

8－1） 巽，酒井：実戦的フロッピーディスク装置入門，産報出版

8－2） 横山英夫：ミニフロッピーディスクドライブのインターフェース，トランジスタ技術，1981年6月号

第9章

9－1） 石田芳：CP/Mの使い方，電子科学シリーズNo 95，産報出版

9－2） インターフェース，CP/M特集号，1981年7月号

索 引

この索引は次の順序で配列してあります。

① 数字ではじまる語（昇順）
② アルファベットではじまる語（ABC順）
③ 和文の語（50音順）

《数 字》

《ABC順》

《あ　行》

《か　行》

《さ　行》

《た　行》

安　居　院　猛

（あぐい　たけし）

1935年生まれ
東京工業大学卒，同博士課程修了
（工学博士）
現在＝東京工業大学工学部像情報
　工学研究施設教授

中　嶋　正　之

（なかじま　まさゆき）

1946年生まれ
東京工業大学卒，同博士課程修了
（工学博士）
現在＝東京工業大学工学部助手

清　弘　智　昭

（きよひろ　ともあき）

1948年生まれ
東京工業大学卒
現在＝山梨大学工学部助手

CP/M手作りマイコン

定価 2,000円

1982年10月15日　初版発行
1983年2月10日　三版発行

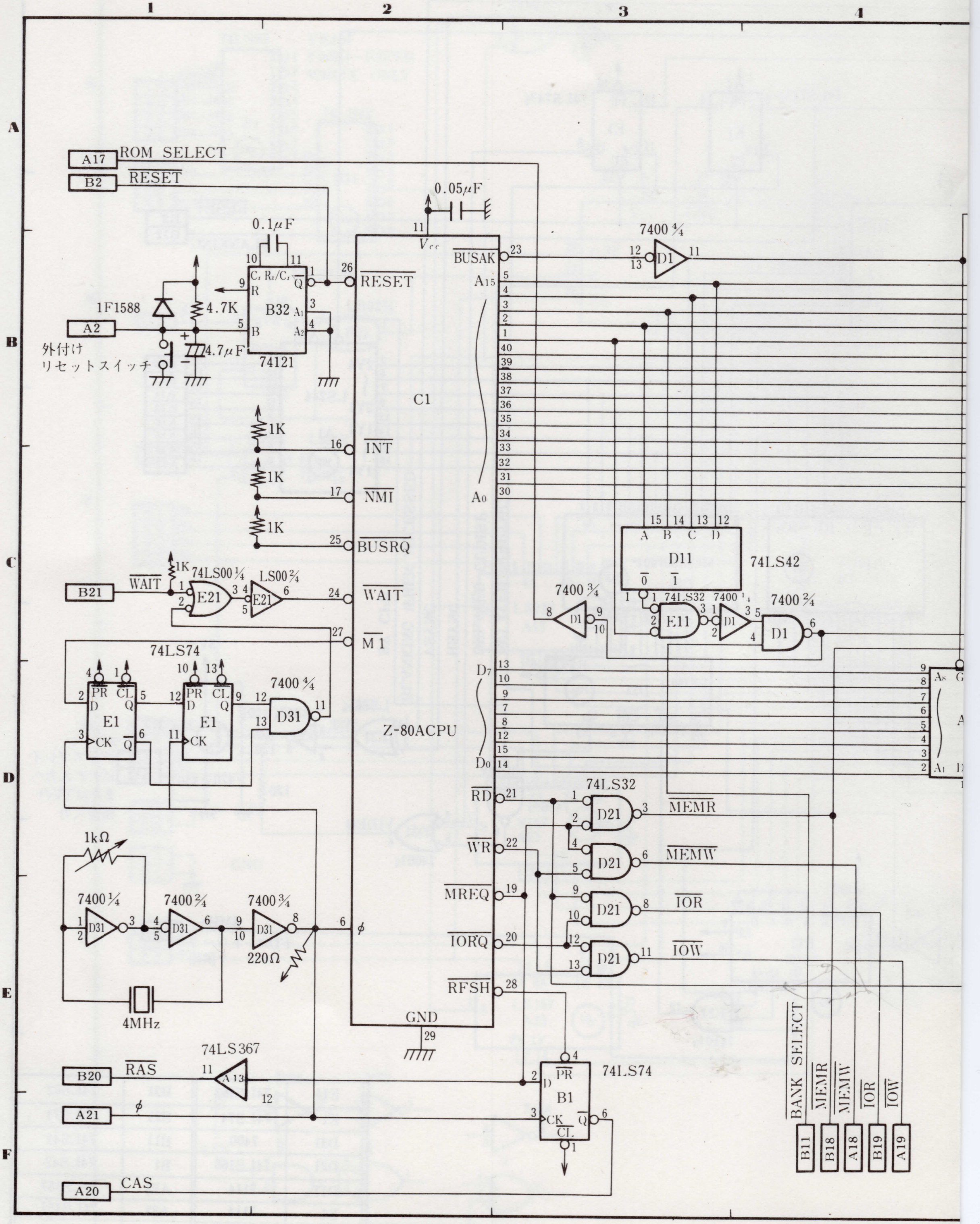
著　者　安　居　院　猛
　　　　中　嶋　正　之
　　　　清　弘　智　昭

発行者　松　山　邦　雄

発行所　産報出版株式会社
東京都台東区東上野4－25－18
郵便番号　110
電話　03(842)0511 大代表
振替　東京1-58204

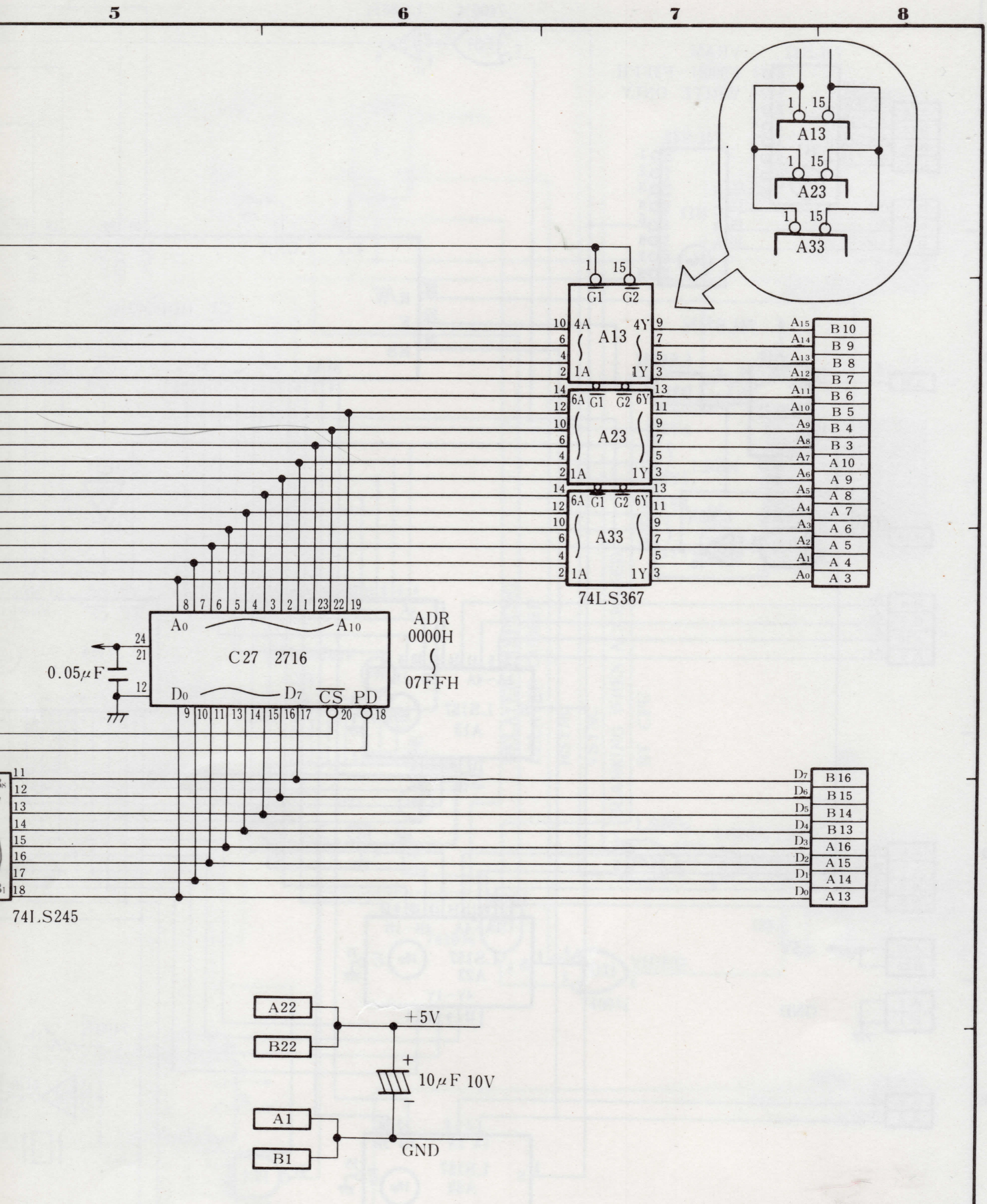
印刷＝株式会社廣済堂
製本＝秋元製本株式会社

万一乱丁，落丁がございましたら，書店または発行所でお取換えいたします。
2055- 133001(03)-2779



1
2
3
4
A
B
C
D
E
F
A17
ROM SELECT
B2
RESET
0.05μF
0.1μF
11
Vcc
BUSAK
23
7400 4/4
D1
26
RESET
A15
A0
1F1588
4.7K
B32
74121
A2
外付け
リセットスイッチ
4.7μF
C1
1K
16
INT
17
NMI
25
BUSRQ
B21
WAIT
74LS00 1/4
LS00 2/4
E21
24
27
M1
74LS74
E1
PR
CL
CK
7400 4/4
D31
Z-80ACPU
D7
D0
D11
74LS42
7400 3/4
74LS32
E11
7400 1/4
7400 2/4
RD
21
WR
22
MREQ
19
IORQ
20
RFSH
28
74LS32
D21
MEMR
MEMW
IOR
IOW
1kΩ
7400 1/4
7400 2/4
7400 3/4
220Ω
φ
4MHz
GND
29
74LS367
B20
RAS
A13
A21
B1
A20
CAS
BANK SELECT
MEMR
MEMW
IOR
IOW
B11
B18
A18
B19
A19

図4.1 CPU基板の



		E11	74LS32	C1	Z-80ACPU
		E1	74LS74	B32	74121
		D31	7400	B1	74LS74
		D21	74LS32	A33	74LS367
		D11	74LS42	A23	74LS367
		D1	7400	A13	74LS367
E21	74LS00	C27	2716	A1	74LS245

折込2

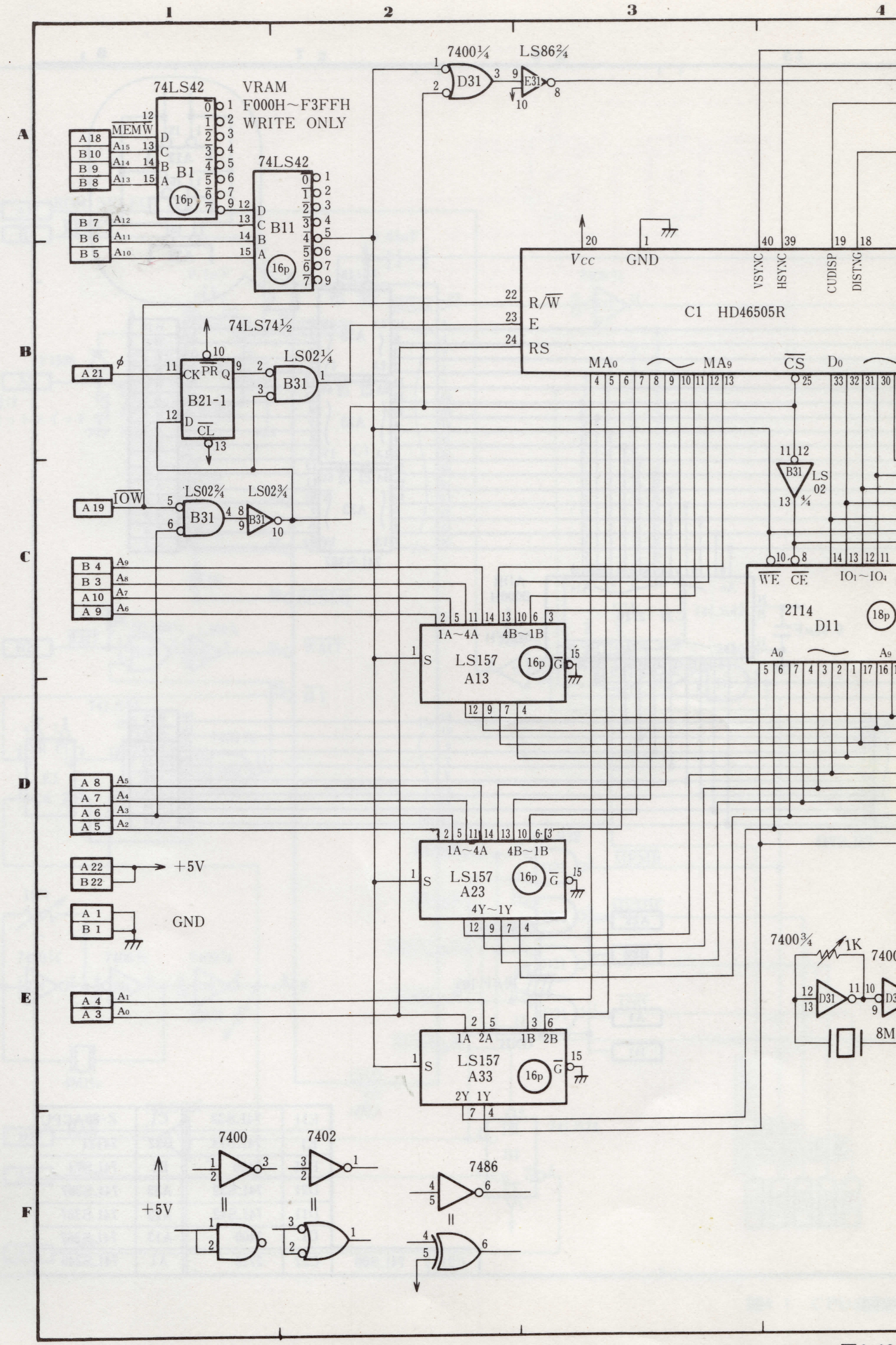
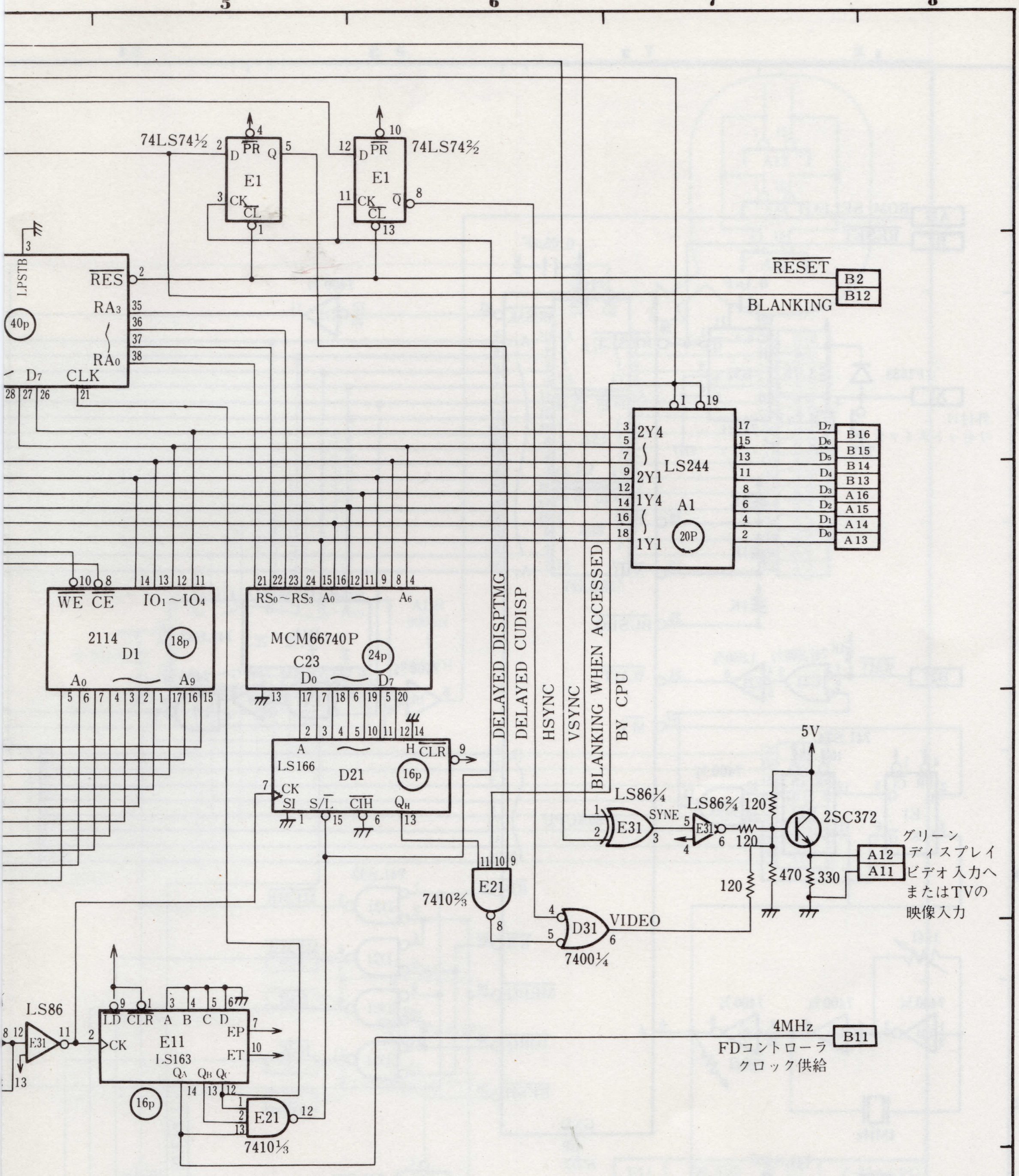

74LS42
VRAM
F000H～F3FFH
WRITE ONLY
74LS42
7400¼
LS86²⁄₄
74LS74½
LS02¼
LS02²⁄₄
LS02¾
C1 HD46505R
2114
D11
LS157
A13
LS157
A23
LS157
A33
+5V
GND
7400
7402
7486
7400¾
1K
8M

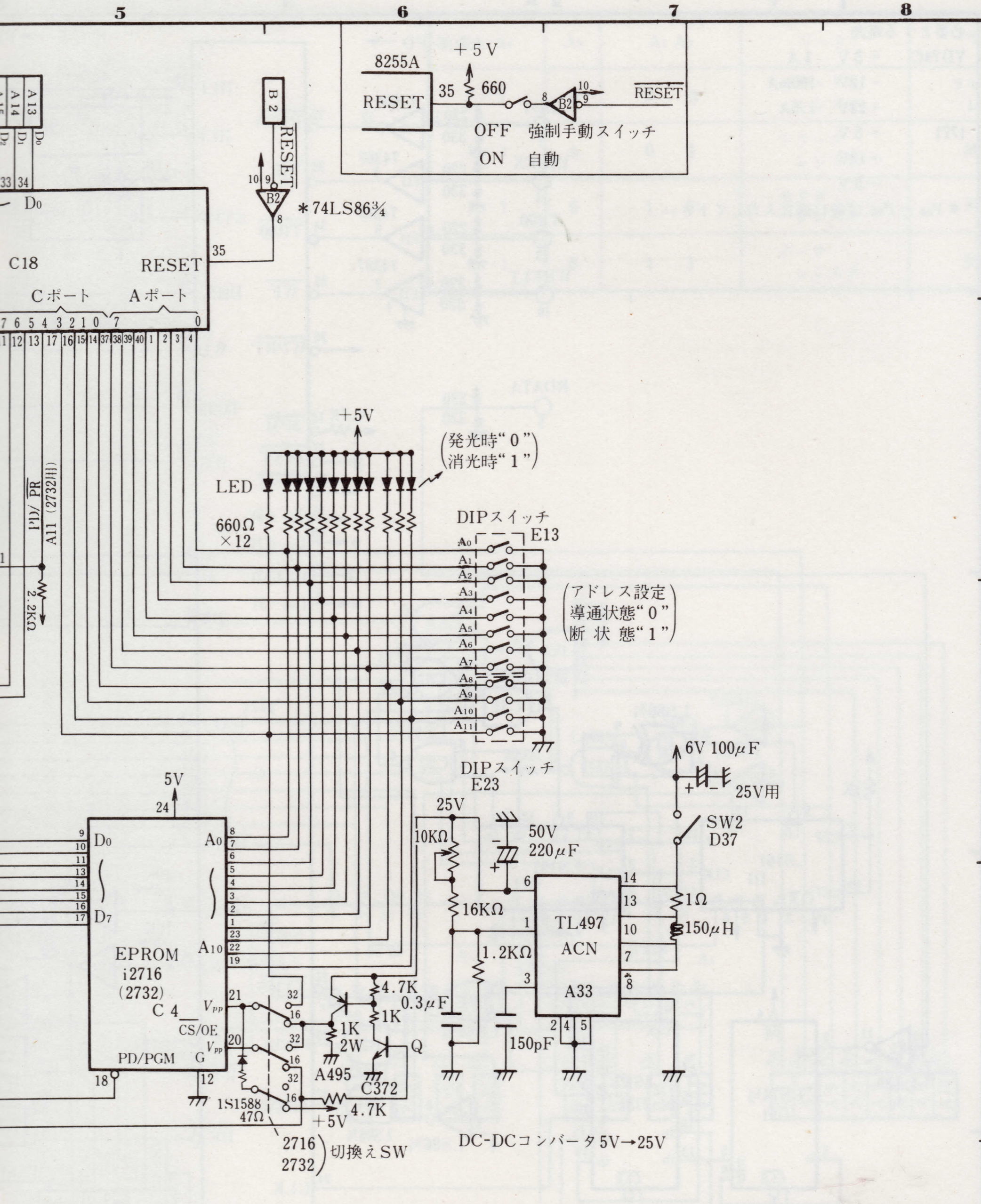
図4.13



E31	74LS86
E21	7410

E11	74LS163	B31	74LS02
E1	74LS74	B21	74LS74
D31	7400	B11	74LS42
D21	74LS166	B1	74LS42
D11	2114	A33	74LS157
D1	2114	A23	74LS157
C23	MCM66740	A13	74LS157
C1	HD46505R	A1	74LS244

CRT制御基板の配線図



A 6	74LS123	D27	74LS245
A15	74LS123	A 2	5V電源SW1
A24	74LS42	D37	25V用SW2
A33	TL497	E 3	DIP SW8P
B 2	74LS86	E13	DIP SW8P
C 4	2716	E23	DIP SW4P
C18	8255A	E48	書込みSW

基板の配線図

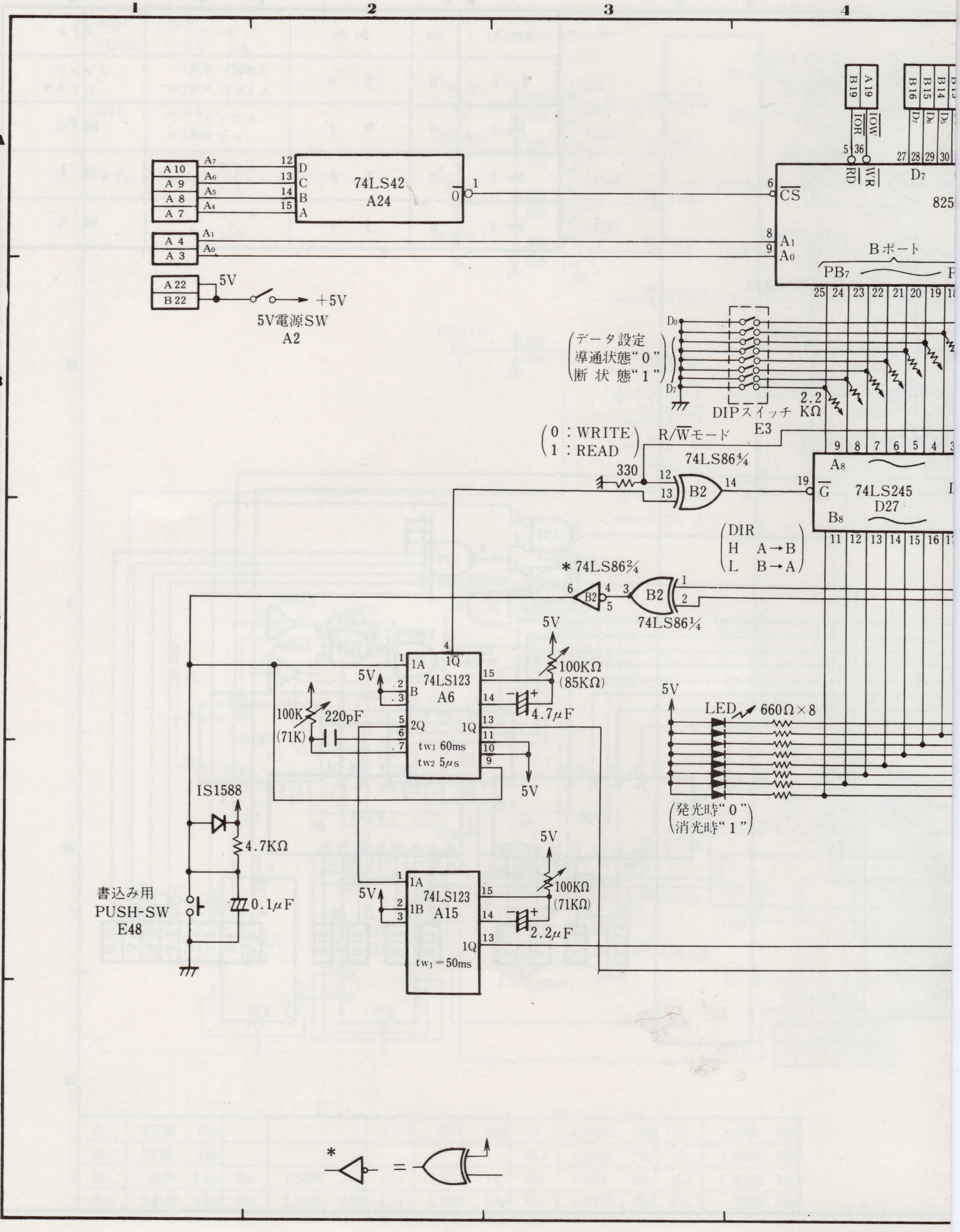
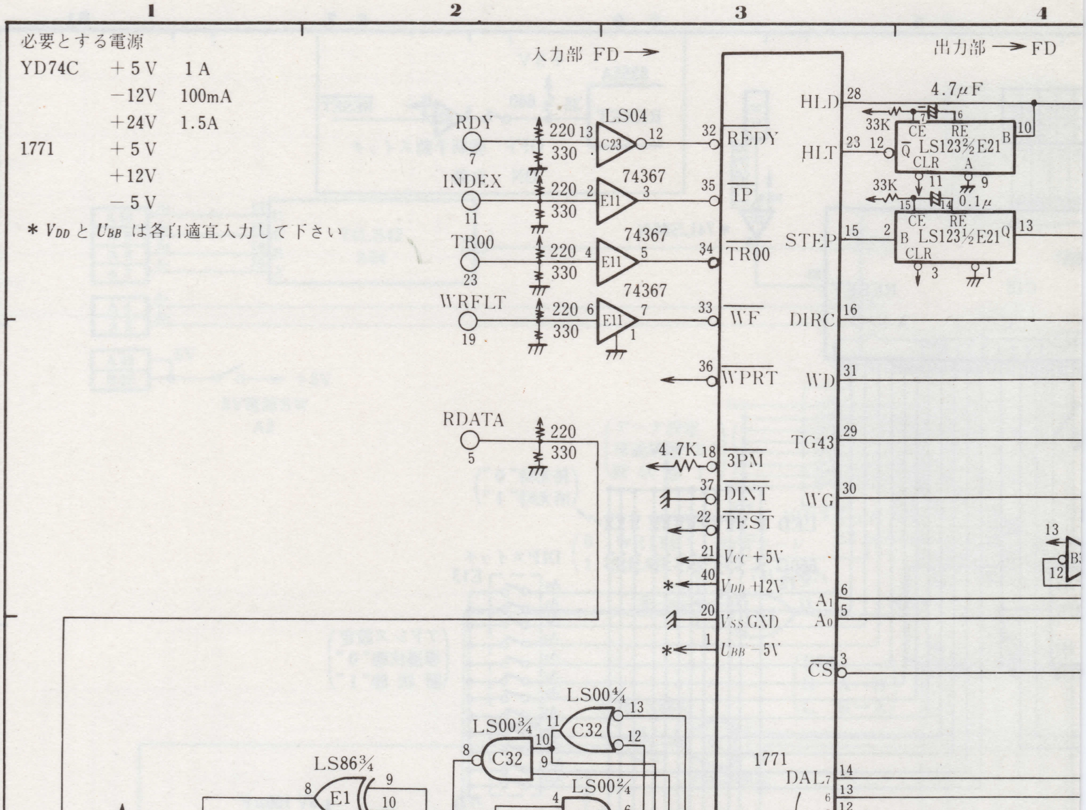

1
2
3
4
A
B
C
D
E
A 10
A 9
A 8
A 7
74LS42
A24
A 4
A 3
A 22
B 22
5V
+5V
5V電源SW
A2
A 19
B 19
IOR
IOW
B16
B15
B14
CS
RD
WR
A1
A0
Bポート
PB7
(データ設定
導通状態"0"
断 状 態"1")
DIPスイッチ
E3
2.2 KΩ
(0 : WRITE
1 : READ)
R/Wモード
330
74LS86 4/4
B2
74LS245
D27
(DIR
H A→B
L B→A)
* 74LS86 2/4
74LS86 1/4
74LS123
A6
100KΩ
(85KΩ)
4.7μF
100K
(71K)
220pF
tw1 60ms
tw2 5μs
5V
LED
660Ω×8
(発光時"0"
消光時"1")
IS1588
4.7KΩ
0.1μF
書込み用
PUSH-SW
E48
74LS123
A15
100KΩ
(71KΩ)
2.2μF
tw1＝50ms

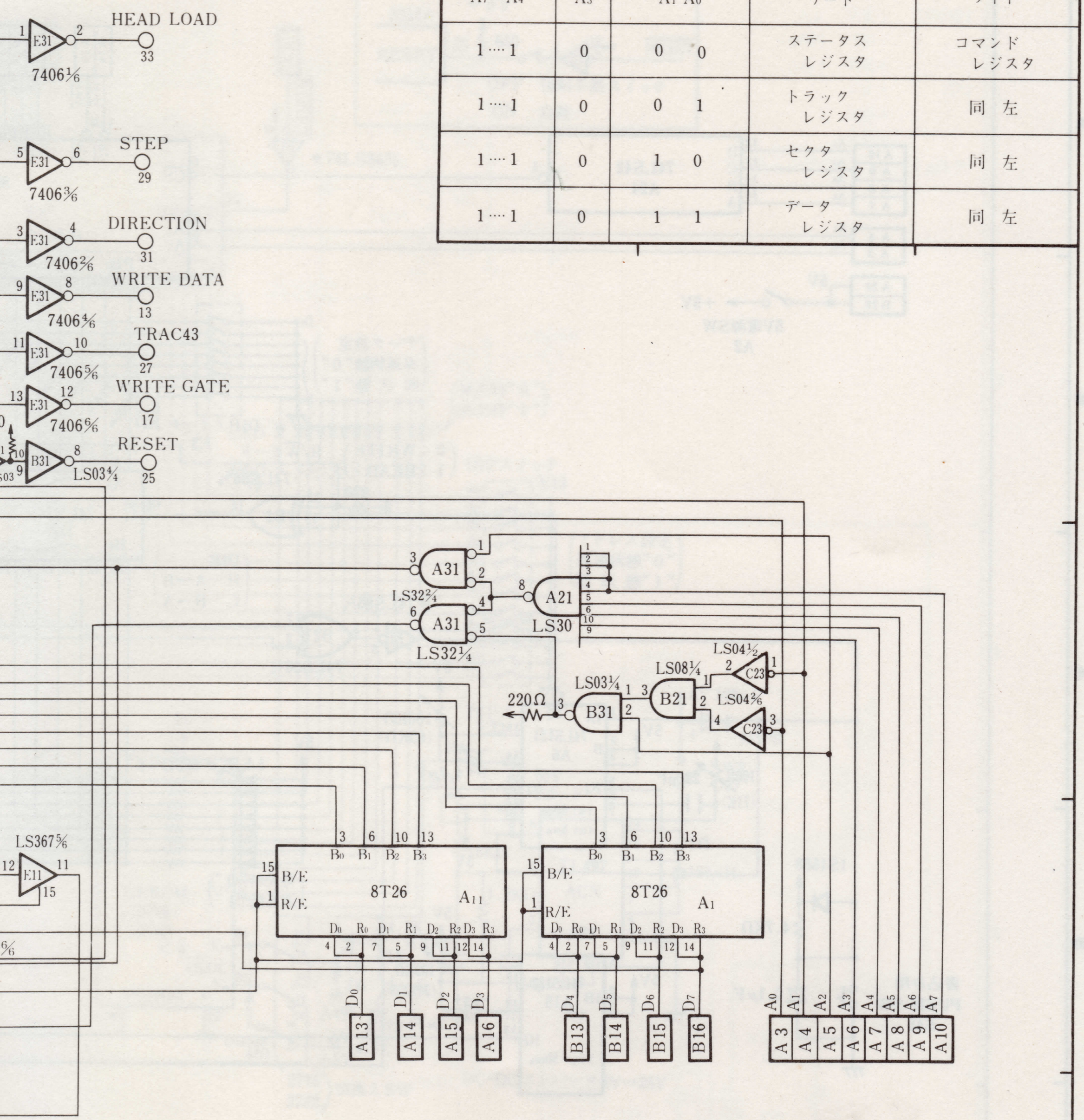
図6.5 EPROMライ



A1	8T26	16p				C1	1771	40p	D1	LS161	16p	E1	LS86	14p
A11	8T26	16p							D11	LS161	16p	E11	LS367	16p
A21	LS30	14p	B21	LS08	14p	C23	LS04	14p	D21	LS74	14p	E21	LS123	14p
A31	LS32	14p	B31	LS03	14p	C32	LS00	14p	D31	LS175	16p	E31	7406	14p

図8.9 フロッピ

A_7～A_4	A_3	A_1 A_0	リード	ライト
1…1	0	0　0	ステータス レジスタ	コマンド レジスタ
1…1	0	0　1	トラック レジスタ	同　左
1…1	0	1　0	セクタ レジスタ	同　左
1…1	0	1　1	データ レジスタ	同　左



ディスク制御基板の配線図

CP/M手作りマイコン